Data Analysis 2- Final Project

Using conditional logistic regressions (Logit Regressions), I will be exploring the relationships that certain explanatory variables (Monthly income/log of monthly income, pitch satisfaction score, number of followups, and whether one has a passport or not) have on the binary dependent variable, whether one purchased the travel package or not. The goal is to see if these variables have an effect on the probability that someone purchased the travel package.

This analysis could be used to understand the factors that have the most effect on whether someonne purchases a travel package, and therefore travel agencies could take the information found in this analysis and derive strategies from it to increase sales.

Topic: Pricing and Sales (Kaggle datatable: Holiday Package Prediction (for the Trip and Travel Company)) https://www.kaggle.com/datasets/susant4learning/holiday-package-purchase-prediction

Variables for study:

- y: ProdTaken
- x: Monthly Income/LnMonthlyIncome
- z(s): PitchSatisfactionScore, NumberOfFollowups, Passport

In [12]:

```
#reading dataset
df = pd.read_csv('https://raw.githubusercontent.com/corinne167225/DataAnalysis2/main/Fin
df.head(3)
```

Out[12]:

	CustomerID	ProdTaken	Age	TypeofContact	CityTier	DurationOfPitch	Occupation	Gender	NumberO
0	200000	1	41.0	Self Enquiry	3	6.0	Salaried	Female	
1	200001	0	49.0	Company Invited	1	14.0	Salaried	Male	
2	200002	1	37.0	Self Enquiry	1	8.0	Free Lancer	Male	

The dataset has 4,888 observations and 20 columns. Each observation represents a customer at the travel agency. *Unit/currency of monthly income is not defined in the dataset or in the explanation of the dataset. Because the values are high in comparison to what I would think if the variable was measured in USD, I assume it is not for the U.S. Therefore, I will refer to the units of monthly income as exactly that: units rather than a specified currency.

I will not be doing a normal regression as the dependent variable (Product Taken) is binary. Between the choices of LPM, Logit, and Probit, I will choose to do Logit regressions because they don't assume linearity, and the range for my dependent variable won't fall outside of [0,1]. This will mean I will need to have a marginal effects table as well for each logit regression as logit regressions don't produce as readily interpretable coefficients as linear probability models (LPMs). I also decided to take Ln of Monthly Income as the numbers were more readily interpretable (also explained later).

In [23]:

```
#marginal effects for Logit Regression (ProdTaken regressed on LnMonthlyIncome)
marginal_effectsxxx = resultxxx.get_margeff()
print(marginal_effectsxxx.summary())
```

```
        Logit Marginal Effects
```

```
=====================================
Dep. Variable:              ProdTaken
Method:                          dydx
At:                           overall
===================================================================================
                     dy/dx    std err          z      P>|z|      [0.025      0.975]
-----------------------------------------------------------------------------------
LnMonthlyIncome    -0.2915      0.043     -6.852      0.000      -0.375      -0.208
===================================================================================
```

Logit Marginal Effects Interpretation: For a 1 unit increase in log monthly income, the probability of someone buying the travel package decreases by about 29%. This coefficient is more easily interpretable than that of the level monthly income variable. This is a big find as log of monthly income seems to have a large effect on whether someone purchases the product or not- interetsingly enough, in the oppsite direction of what one would expect. This could mean other variables have something to do with this effect. Will use LnMonthlyIncome in the rest of the regressions. The coefficient is statistically significant and can therefore be applied to the population.

In [28]:

```
#marginal effects for Logit Regression (ProdTaken regressed on LnMonthlyIncome, PitchSat
marginal_effects2 = result2.get_margeff()
print(marginal_effects2.summary())
```

```
        Logit Marginal Effects
=====================================
Dep. Variable:              ProdTaken
Method:                          dydx
At:                           overall
========================================================================================
==
                            dy/dx    std err          z      P>|z|      [0.025      0.97
5]
----------------------------------------------------------------------------------------
--
LnMonthlyIncome           -0.2926      0.043     -6.871      0.000      -0.376      -0.2
09
PitchSatisfactionScore     0.0167      0.004      3.883      0.000       0.008       0.0
25
========================================================================================
==
```

Logit Marginal Effects Interpretation: Holding pitch satisfaction score constant, a one unit increase in log monthly income means that on average, the probability of someone purchasing the travel package decreases by about 29%. Holding log of monthly income constant, a one unit increase in pitch satisfaction score on average, means that the probability of someone purchasing the travel package increases by 1.67%. So indeed it seems true that the pitch satisfaction score doesn't have much to do with if one purchases a package or not, at least in comparison to the log of monthly income. All coefficients are statistically significant and can therefore be applied to the population.

In [31]:

```
#marginal effects for Logit Regression (ProdTaken regressed on LnMonthlyIncome, PitchSat
#Followups)
marginal_effects3 = result3.get_margeff()
print(marginal_effects3.summary())
```

```
        Logit Marginal Effects
=====================================
Dep. Variable:              ProdTaken
Method:                          dydx
At:                           overall
========================================================================================
==
                            dy/dx    std err          z      P>|z|      [0.025      0.97
5]
------------------------------------------------------------------------------
--
LnMonthlyIncome           -0.3751      0.048     -7.782      0.000     -0.470     -0.2
81
PitchSatisfactionScore     0.0171      0.004      4.068      0.000      0.009      0.0
25
NumberOfFollowups          0.0569      0.007      8.623      0.000      0.044      0.0
70
==============================================================================
==
```

Logit Marginal Effects Interpretation: Holding pitch satisfaction score and number of followups constant, a one unit increase in log monthly income means that on average, the probability of someone purchasing the travel package decreases by about 37%. Holding log of monthly income and number of followups constant, a one unit increase in pitch satisfaction score on average, means that the probability of someone purchasing the travel package increases by 1.71%. Holding ln monthly income and pitch satisfaction score constant, an one unit increase in number of followups, on average, means the probability of someome purchasing the travel package increases by 5.69%. All coefficients are statistically significant and can therefore be applied to the population.

In [34]:

```
#marginal effects for Logit Regression (ProdTaken regressed on LnMonthlyIncome, PitchSat
#Followups, Passport)
marginal_effects4 = result4.get_margeff()
print(marginal_effects4.summary())
#maybe Passport is one of the largest factors behind why people buy these packages... sa
#or can't...if you have passport, probability of purchasing package increases by 18.93 p
```

```
        Logit Marginal Effects       
=====================================
Dep. Variable:              ProdTaken
Method:                          dydx
At:                           overall
==============================================================================
==
                            dy/dx    std err          z      P>|z|      [0.025      0.97
5]
------------------------------------------------------------------------------
--
LnMonthlyIncome           -0.3677      0.046     -7.936      0.000     -0.458     -0.2
77
PitchSatisfactionScore     0.0170      0.004      4.264      0.000      0.009      0.0
25
NumberOfFollowups          0.0542      0.006      8.660      0.000      0.042      0.0
66
Passport                   0.1952      0.010     19.539      0.000      0.176      0.2
15
==============================================================================
==
```

Logit Marginal Effects Interpretation: Holding pitch satisfaction score, number of followups, and having/not having a passport constant, a one unit increase in log monthly income means that on average, the probability of someone purchasing the travel package decreases by about 36%. Holding log of monthly income, number of followups, and having/no having a passport constant, a one unit increase in pitch satisfaction score on average, means that the probability of someone purchasing the travel package increases by 1.70%. Holding ln monthly income, pitch satisfaction score, and having/not having a passport constant, a one unit increase in number of followups, on average, means the probability of someome purchasing the travel package increases by 5.42% Holding ln monthly income, pitch satisfaction score, and number of followups constant, having a passport, on average, means the probability of someone purchasing a travel package increases by 19.52% This is a big find as it shows those without passports are almost 20% less likely to purchase the travel package. If the packages were for international travel destinations, this would make sense. Maybe the customers would not be able to get their passports in time, or didn't want to go through the hassle of doing so. If the packages for the most part were domestic destinations, this would perhaps have less meaning. All coefficients are statistically significant and can therefore be applied to the population.

SEEN IN APPENDIX: The variable of passport seems to provide the best fit to the model (highest R-squared, highest Log-loss, lowest Brier-score) which makes sense to due the effect we saw in the marginal effects table involving passport. Interestingly LnIncome (meaning LnMonthlyIncome) has one of the worst fits of the variables. This confirms what was earlier believed: LnMonthlyIncome on its own doesn't seem to have a strong relationship with whether one purchased the travel package or not.

Summary:

All of the variables seem to have an effect on the probability that someone purchases a travel package due to the p-values being near 0 and the confidence intervals not including 0. However, some seem to have more effects than others. Based on the measures of fit, having a passport seems to have the most significant effect on probability of purchasing a travel package, followed by number of followups, then pitch satisfaction score. Lastly, the log of monthly income seems to have the smallest effect on the probability that someone purchased the travel package.

Causal Interpretation: For the most part you should never conclude that the explanatory variables cause the dependent variable, however, in this instance one can say that using these logit regressions, as we add more controlled variables to the regression, we get closer to understanding the possible causal interpretation.

For this travel agency (The Trip and Travel Company), I would suggest that they encourage more of their customers to get their passports, or perhaps focus on domestic destinations rather than international locations, as I believe this would help their travel package sales. I would also advise to shoot for 3 followups minimum and 4 followups maximum to help increase the likelihood that their customers actually purchase the travel packages. Furthermore, the travel agency should perhaps focus less on a target audience of richer clients, but perhaps those who tend to have a lower monthly income to increase likelihood of selling packages.

APPENDIX

Because the main report had to be maximum 4 pages, a lot of the above code and cells may seem as if they came from nothing, however, all the code below is was previously intertwined with that above at various points. Below you can see the imported libraries, data cleaning, Logit regression for Monthly Income before I decided to ransform the variable to Log of Monthly Income, various scripts for aggregations to be used in the visualizations, visualizations for each variable and their interpretations, the Logit regressions for each added explanatory variable, and the predicted probabilities of each variable that were used to create the measures of fit.

For a more concise idea of the process and ORDER in which things actually are, please see `Final_Project0.ipynb`

In [11]:

```
#importing libraries
import warnings

import pandas as pd
from plotnine import *
import statsmodels.api as sm
import statsmodels.formula.api as smf
from stargazer.stargazer import Stargazer
import numpy as np
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.metrics import log_loss
warnings.filterwarnings("ignore")
```

In [13]:

```
#Data Cleaning
#changing NaN in age, monthlyincome, and number of followups to be 0 so it won't affect
df['Age'].fillna(0, inplace = True)
df['MonthlyIncome'].fillna(0, inplace = True)
df['NumberOfFollowups'].fillna(0, inplace = True)
df['DurationOfPitch'].fillna(0, inplace = True)
#Count column to help with aggregations
df['Count'] = 1
df
```

Out[13]:

	CustomerID	ProdTaken	Age	TypeofContact	CityTier	DurationOfPitch	Occupation	Gender	Numb
0	200000	1	41.0	Self Enquiry	3	6.0	Salaried	Female	
1	200001	0	49.0	Company Invited	1	14.0	Salaried	Male	
2	200002	1	37.0	Self Enquiry	1	8.0	Free Lancer	Male	
3	200003	0	33.0	Company Invited	1	9.0	Salaried	Female	
4	200004	0	0.0	Self Enquiry	1	8.0	Small Business	Male	
...	...	...	...	...	...	...	...	...	
4883	204883	1	49.0	Self Enquiry	3	9.0	Small Business	Male	
4884	204884	1	28.0	Company Invited	1	31.0	Salaried	Male	
4885	204885	1	52.0	Self Enquiry	3	17.0	Salaried	Female	
4886	204886	1	19.0	Self Enquiry	3	16.0	Small Business	Male	
4887	204887	1	36.0	Self Enquiry	1	14.0	Salaried	Male	

4888 rows × 21 columns

In [14]:

```
#description of monthly income for those who didn't take package
df.loc[df['ProdTaken'] == 0]['MonthlyIncome'].describe()
#surprised that mean is higher here... BUT is it statistically significant???
```

Out[14]:

```
count     3968.000000
mean     22747.214466
std       7494.701528
min          0.000000
25%      20130.750000
50%      22413.500000
75%      25760.250000
```

```
max        98678.000000
Name: MonthlyIncome, dtype: float64
```

In [15]:

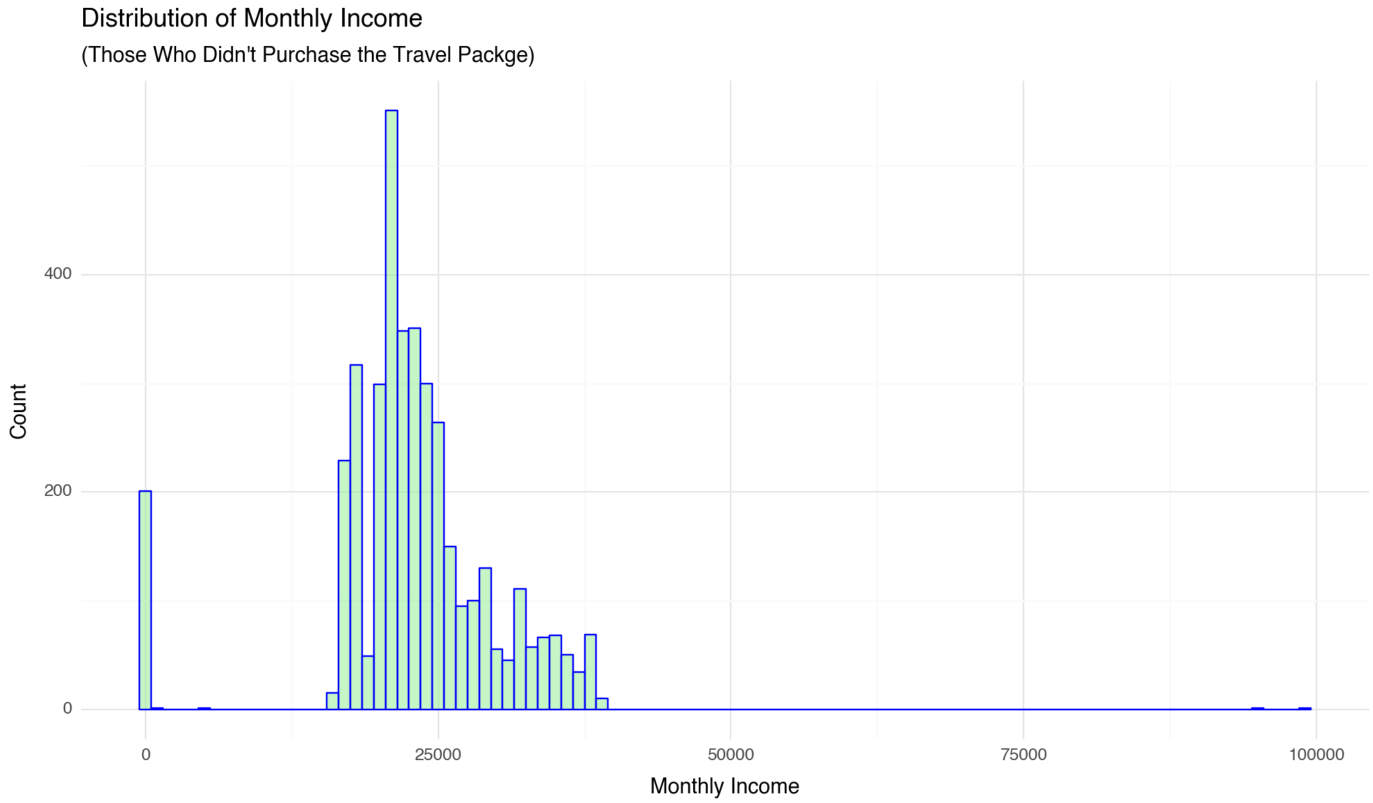
```
#description of monthly income for those who did take package
df.loc[df['ProdTaken'] == 1]['MonthlyIncome'].describe()
```

Out[15]:

```
count      920.000000
mean     21401.598913
std       6114.176598
min          0.000000
25%      17951.000000
50%      21095.500000
75%      23857.500000
max      38537.000000
Name: MonthlyIncome, dtype: float64
```

In [16]:

```
#histogram to see monthly income distribution for those who didn't take package
df_no_package = df[df['ProdTaken'] == 0]
hist1 = (ggplot(df_no_package, aes(x = 'MonthlyIncome'))
+ geom_histogram(binwidth = 1000, fill = 'lightgreen', color = 'blue', alpha = 0.5)
+ labs(x = 'Monthly Income', y = 'Count', title = "Distribution of Monthly Income",
       subtitle = "(Those Who Didn't Purchase the Travel Packge)")
+ theme_minimal() + theme(figure_size=(10, 6)))
hist1
```

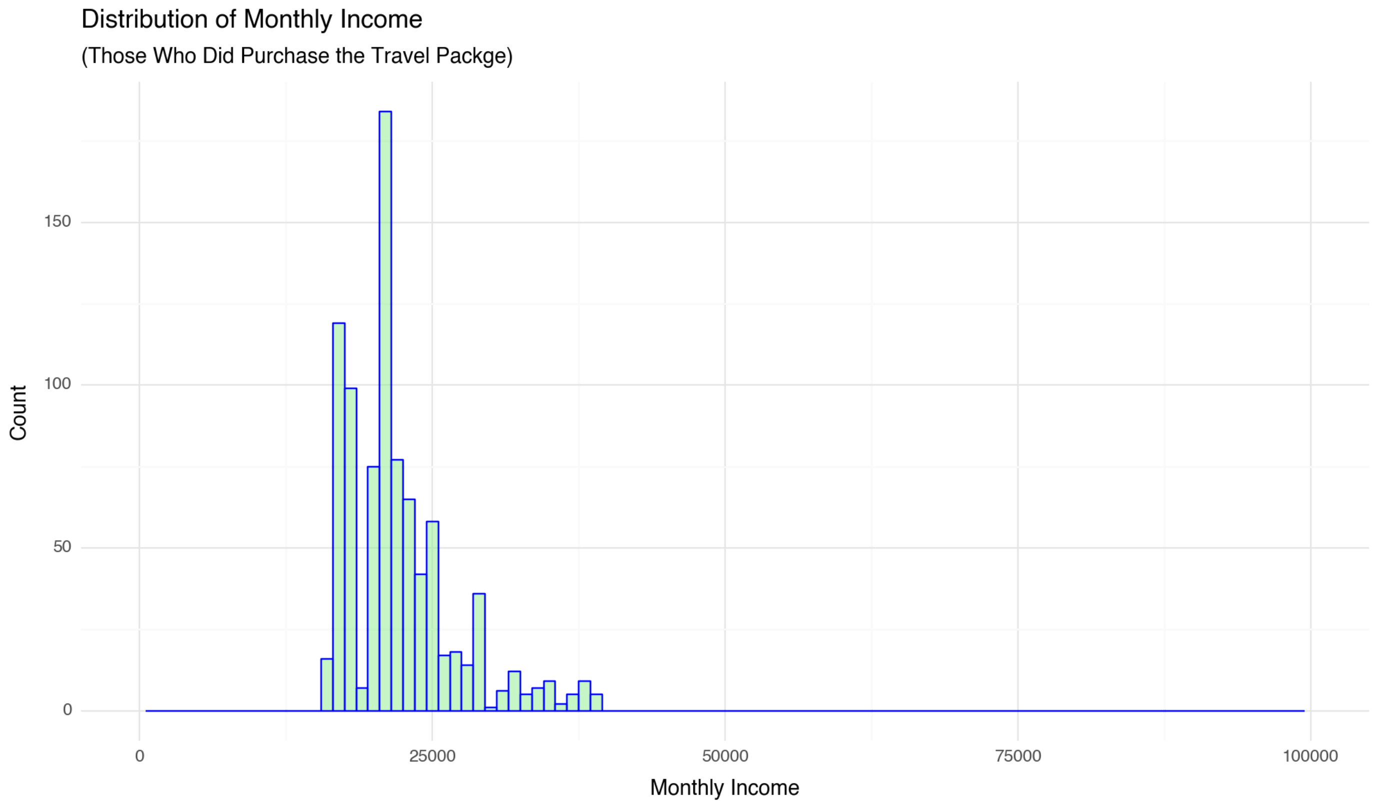


Out[16]:

```
<Figure Size: (1000 x 600)>
```

In [17]:

```
#histogram to see monthly income distribution for those who did take package
df_package = df[df['ProdTaken'] == 1]
hist2 = (ggplot(df_package, aes(x = 'MonthlyIncome'))
+ geom_histogram(binwidth = 1000, fill = 'lightgreen', color = 'blue', alpha = 0.5)
+ labs(x = 'Monthly Income', y = 'Count', title = "Distribution of Monthly Income",
       subtitle = "(Those Who Did Purchase the Travel Packge)")
+ theme_minimal() + theme(figure_size=(10, 6))
+ scale_x_continuous(limits=(0, 100000)))
hist2
```



```
Out[17]: <Figure Size: (1000 x 600)>
```

The histograms are pretty similar although 1st histogram seems to be more disributed towards center and have more variation in x (both lower and larger values shown)- however, this is most likely due to the fact that a larger proportion of customers did not buy the travel package.

```
In [18]: #Logit regression of ProdTaken on MonthlyIncome
x_vars = sm.add_constant(df[['MonthlyIncome']])
model = sm.Logit(df['ProdTaken'], x_vars)
result1 = model.fit(cov_type = 'HC1')
```

```
Optimization terminated successfully.
         Current function value: 0.481050
         Iterations 5
```

```
In [19]: #marginal effects for logit regression (ProdTaken regressed on MonthlyIncome)
marginal_effects = result1.get_margeff()
print(marginal_effects.summary())
```

```
        Logit Marginal Effects
=====================================
Dep. Variable:              ProdTaken
Method:                          dydx
At:                           overall
=================================================================================
                   dy/dx    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
MonthlyIncome -3.779e-06    6.6e-07     -5.730      0.000   -5.07e-06   -2.49e-06
=================================================================================
```

Logit Marginal Effects Interpretation: For a one unit increase in monthly income, a person is on average 3.779e-06 times less likely to buy the travel package. The coefficient is statistically significant. This interpretation is not the nicest, so we will also look at the log of monthly income to see if that produces more interpretable coefficients.

- The coefficient is statistically significant and can therefore be applied to the population.

In [20]:

```
#adding a log of monthly income comlumn represented as 'LnMonthlyIncome' to try to produ
df['LnMonthlyIncome'] = np.log(df['MonthlyIncome'])
```

In [21]:

```
#seeing how many nan values for LnMonthlyIncome and replacing with 0
df['LnMonthlyIncome'].fillna(0, inplace = True)
df['LnMonthlyIncome'].isna().sum()
#seeing how many infinite values in LnMonthlyIncome and replacing them
df['LnMonthlyIncome'].replace([np.inf, -np.inf], np.nan, inplace=True)
df.dropna(inplace=True)
df['LnMonthlyIncome']
```

Out[21]:

```
0        9.951944
1        9.909967
2        9.746249
3        9.793059
4        9.823795
           ...
4883    10.187764
4884     9.962322
4885    10.367850
4886     9.917834
4887    10.087516
Name: LnMonthlyIncome, Length: 4423, dtype: float64
```

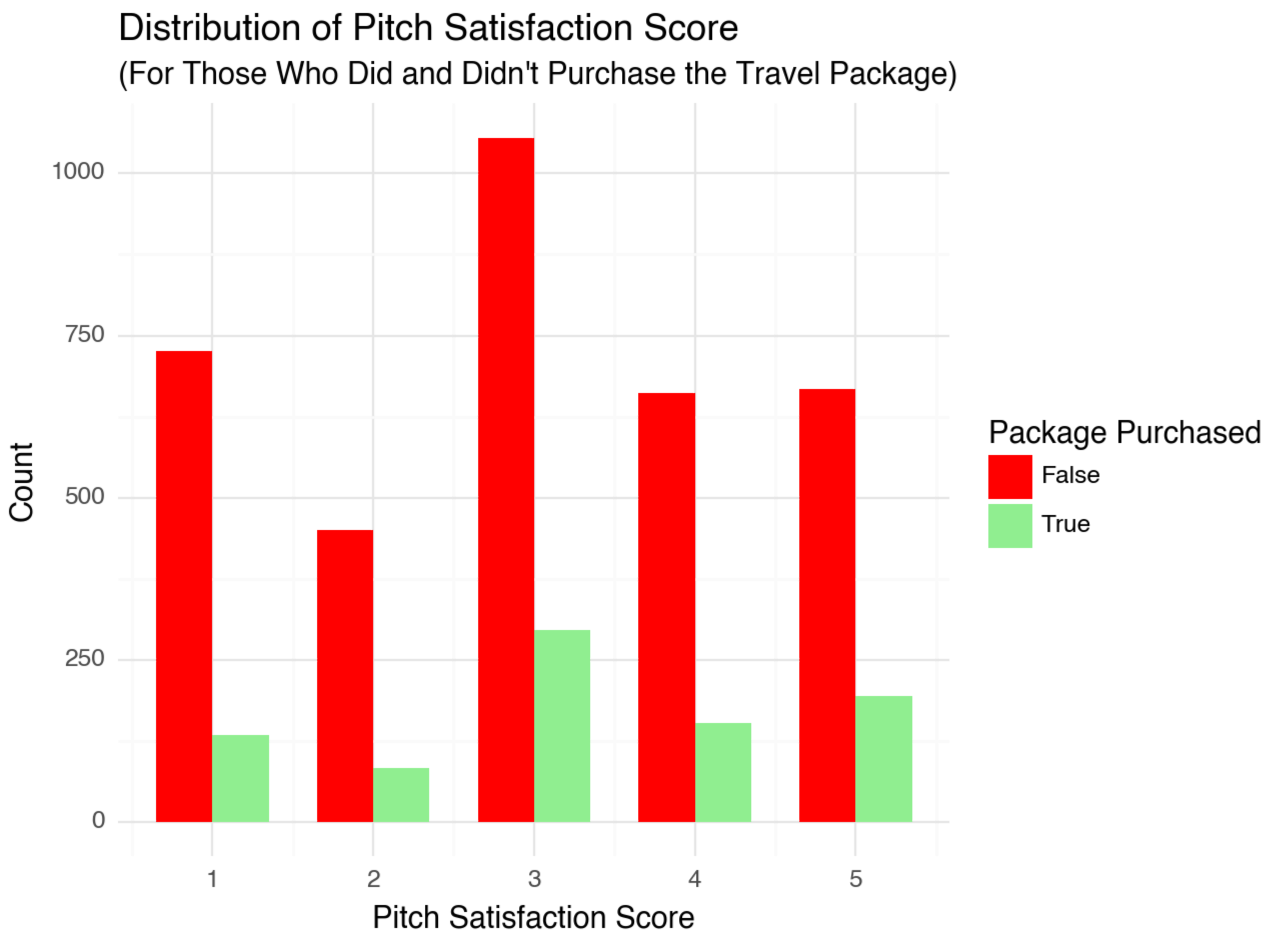
In [22]:

```
#trying log monthly income to see if numbers are easier to interpret
#Logit regression of ProdTaken on LnMonthlyIncome
x_varsxxx = sm.add_constant(df['LnMonthlyIncome'])
modelxxx = sm.Logit(df['ProdTaken'], x_varsxxx)
resultxxx = modelxxx.fit(cov_type = 'HC1')
```

```
Optimization terminated successfully.
         Current function value: 0.484659
         Iterations 6
```

In [24]:

```
#pitch satisfaction score
#grouped bar chart, 1 bar = not taken, 1 bar = taken
group_bar1 = (ggplot(df, aes(x='PitchSatisfactionScore', fill= (df['ProdTaken'] ==1)))
+ geom_bar(position='dodge', width=0.7)
+ scale_fill_manual(values=['red', 'lightgreen'])
+ labs(x='Pitch Satisfaction Score', y='Count', fill='Package Purchased', title = 'Distr
subtitle = "(For Those Who Did and Didn't Purchase the Travel Package)")
+ theme_minimal())
group_bar1
```



Out[24]:

```
<Figure Size: (640 x 480)>
```

It seems the for those who did purchase the travel package, the most purchased it only with a pitch satisfaction score of 3 which is very interesting. Possible reasons for this are that maybe those who purchased the package were planning on going on a vacation and getting the package regardless of how well it was sold to them or not.

Overall it seems that the average pitch satisfaction score was 3, both for those who did and didn't purchase the package.

In [25]:

```
#want to see how many customer didn't and did purchase package
df['ProdTaken'].value_counts()
```

Out[25]:

```
ProdTaken
0    3560
1     863
Name: count, dtype: int64
```

It should be noted that many more people did not take the product. I would have expected more who did not take product to be higher in lower pitch satisfaction scores and those who did take product to be much higher in higher satisfaction scores. There can be many reasons for this, one possibly being that the effect of pitch satisfaction score on whether one purchased the package or not was minimal.

In [26]:

```
#Logit regression for Product Taken on LnMonthlyIncome and PitchSatisfactionScore
x_vars = sm.add_constant(df[['LnMonthlyIncome', 'PitchSatisfactionScore']])
model2 = sm.Logit(df['ProdTaken'], x_vars)
result2 = model2.fit(cov_type = 'HC1')
```

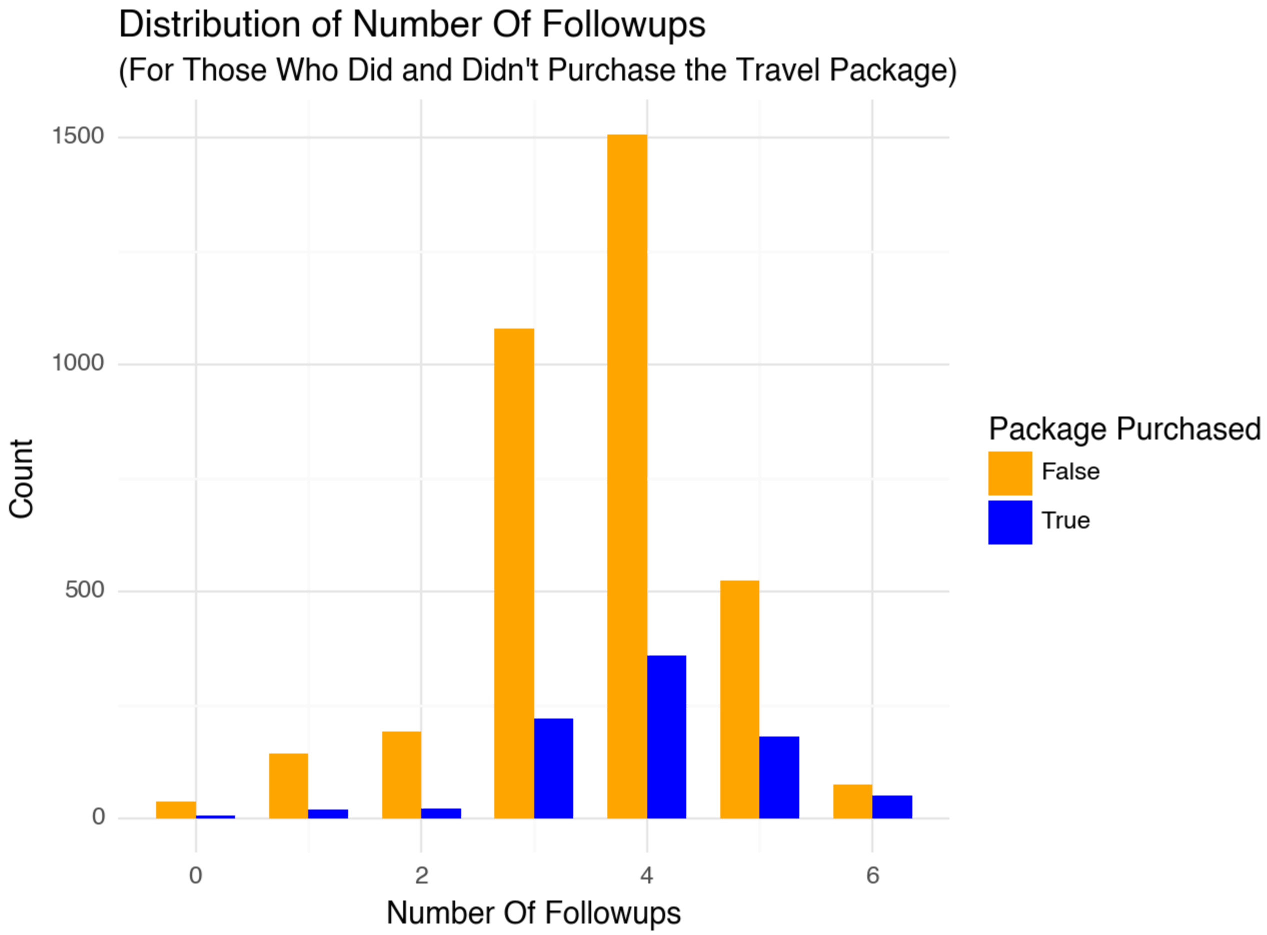
```
Optimization terminated successfully.
         Current function value: 0.482997
         Iterations 6
```

In [27]:

```
#for possible future needed aggregations, creating 'SumCount column'
df_package['SumCount'] = df_package['Count'].sum()
```

In [29]:

```
#numberoffollowups visualization
group_bar2 = (ggplot(df, aes(x='NumberOfFollowups', fill= (df['ProdTaken'] ==1)))
+ geom_bar(position='dodge', width=0.7)
+ scale_fill_manual(values=['orange', 'blue'])
+ labs(x='Number Of Followups', y='Count', fill='Package Purchased', title = 'Distributi
subtitle = "(For Those Who Did and Didn't Purchase the Travel Package)")
+ theme_minimal())
group_bar2
```



Out[29]:

```
<Figure Size: (640 x 480)>
```

Seems that for both those who did and didn't purchase the packages, the aveerage number of followups was between 3 and 4. It is interesting because because at the 3rd followup, it seems that MANY more people decided not to buy the package in comparison to the number who didn't buy at 2 followups.

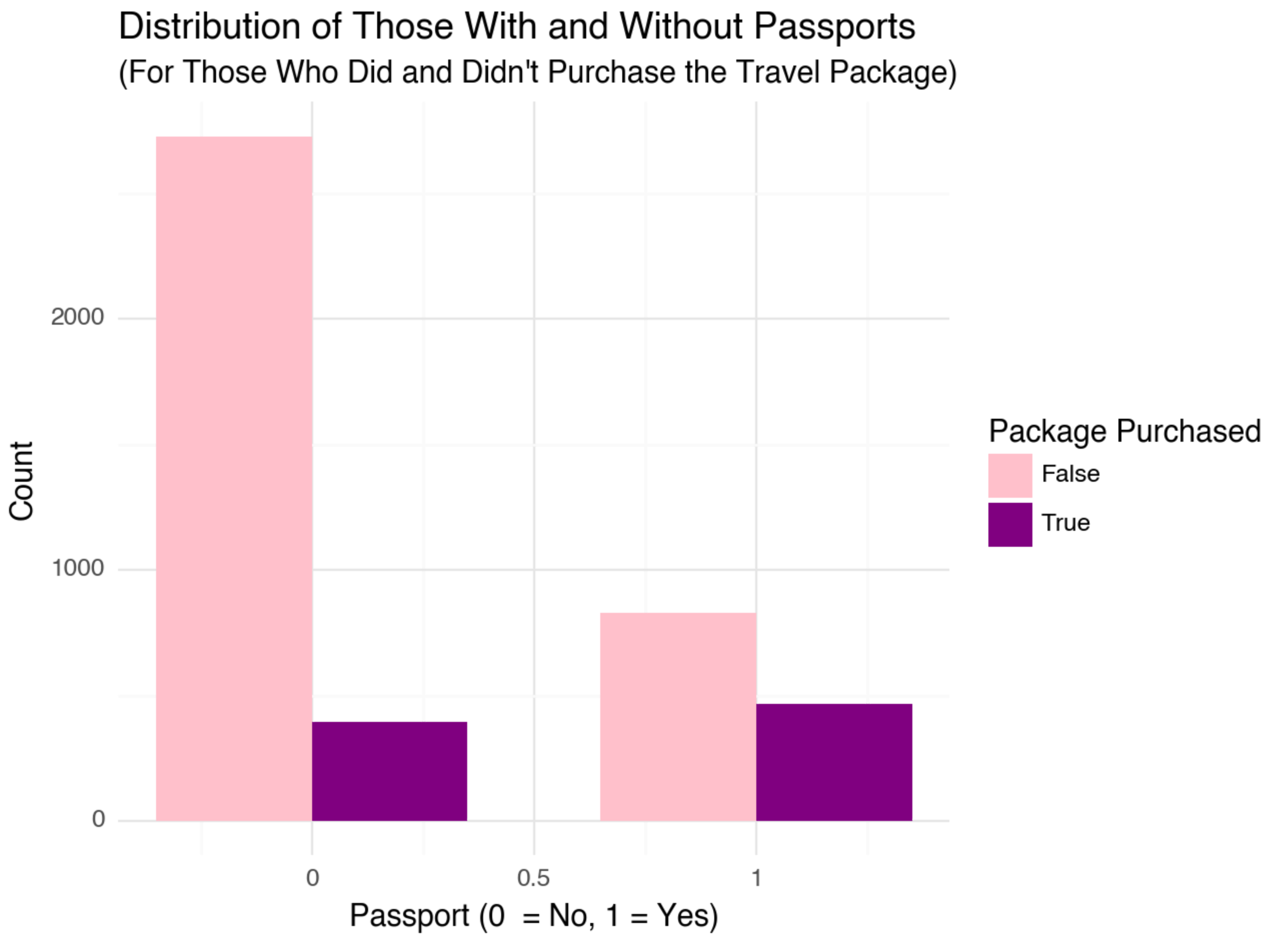
In [30]:

```
#Logit regression of ProdTaken on LnMonthlyIncome, PitchSatisfactionScore, and NumberOfF
x_vars = sm.add_constant(df[['LnMonthlyIncome', 'PitchSatisfactionScore', 'NumberOfFollo
model3 = sm.Logit(df['ProdTaken'], x_vars)
result3 = model3.fit(cov_type = 'HC1')
```

```
Optimization terminated successfully.
         Current function value: 0.472384
         Iterations 6
```

In [32]:

```
#passport visualization
group_bar3 = (ggplot(df, aes(x="Passport", fill= (df['ProdTaken'] ==1)))
+ geom_bar(position='dodge', width=0.7)
+ scale_fill_manual(values=['pink', 'purple'])
+ labs(x='Passport (0  = No, 1 = Yes)', y='Count', fill='Package Purchased', title = 'Di
subtitle = "(For Those Who Did and Didn't Purchase the Travel Package)")
+ theme_minimal())
group_bar3
#interesting because shows that there were more people without passports who didn't purc
#also, there were more withh passports who did purchase than those without who did ----
```



Out[32]:

```
<Figure Size: (640 x 480)>
```

More people who purchased the travel package had their passports (by a minimal amount). More people who did not purchase the package did not have their passports. This is interesting as it might have something to say about the effect having a passport or not had on whether people bought a travel package (if statistically significant).

In [33]:

```
#Logit regression of ProdTaken on LnMonthlyIncome, PitchSatisfactionScore, NumberOfFollo
x_vars = sm.add_constant(df[['LnMonthlyIncome', 'PitchSatisfactionScore', 'NumberOfFollo
model4 = sm.Logit(df['ProdTaken'], x_vars)
result4 = model4.fit(cov_type = 'HC1')
#has the highest pseudo R-squared with value of 0.1120
```

```
Optimization terminated successfully.
         Current function value: 0.438287
         Iterations 6
```

In [35]:

```
#creating predicted probabilities for each explanatory variable
df["PredLogitLnIncome"] = resultxxx.predict()
df["PredLogitPitch"] = result2.predict()
df["PredLogitFollowup"] = result3.predict()
df["PredLogitPassport"] = result4.predict()
```

In [36]:

```
#Fit of predicted probabilities:

fit_predicted_prob_df = pd.DataFrame(
    {
        "R-squared": [
            r2_score(df["ProdTaken"], df["PredLogitLnIncome"]),
            r2_score(df["ProdTaken"], df["PredLogitPitch"]),
            r2_score(df["ProdTaken"], df["PredLogitFollowup"]),
            r2_score(df["ProdTaken"], df["PredLogitPassport"])

        ],
        "Brier-score": [
            mean_squared_error(df["ProdTaken"], df["PredLogitLnIncome"]),
            mean_squared_error(df["ProdTaken"], df["PredLogitPitch"]),
            mean_squared_error(df["ProdTaken"], df["PredLogitFollowup"]),
            mean_squared_error(df["ProdTaken"], df["PredLogitPassport"])

        ],
        "Log-loss": [
            -1 * log_loss(df["ProdTaken"], df["PredLogitLnIncome"]),
            -1 * log_loss(df["ProdTaken"], df["PredLogitPitch"]),
            -1 * log_loss(df["ProdTaken"], df["PredLogitFollowup"]),
            -1 * log_loss(df["ProdTaken"], df["PredLogitPassport"])
        ],



    }, index=["LnIncome", "Pitch", "Followup", "Passport"],).T.round(3)
```

In [37]:

```
#fit of predicted probabilites
fit_predicted_prob_df
```

Out[37]:

	LnIncome	Pitch	Followup	Passport
R-squared	0.019	0.022	0.049	0.134
Brier-score	0.154	0.154	0.149	0.136
Log-loss	-0.485	-0.483	-0.472	-0.438

In [ ]:

```
#converting/'knitting' Jp Nb to pdf
!jupyter nbconvert --to webpdf Final_Project-Copy2.ipynb
```

In [ ]: