The in-depth analysis on customer behaviour and change in the metrics after the integration of new Package plan. Nurly Kuzdikbay for Beeline

In [26]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from functools import reduce
from scipy.stats import ttest_ind
```

In [17]:
```
df = pd.read_excel('Данные_кейс.xlsx')

# Convert dates to datetime objects
df['Дата активации абонента'] = pd.to_datetime(df['Дата активации абонента'], dayfirst=True)
df.head(5)
```

Out[17]:

	Код абонента	Регион	Дата активации абонента	2016-01-01 Расход абонента (тенге)	2016-02-01 Расход абонента (тенге)	2016-03-01 Расход абонента (тенге)	2016-04-01 Расход абонента (тенге)	2016-05-01 Расход абонента (тенге)	2016-06-01 Расход абонента (тенге)	2016-07-01 Расход абонента (тенге)	...	2016-01-01 тип тарифного плана	2016-02-01 тип тарифного плана	2016-03-01 тип тарифного плана	2016-04-01 тип тарифного плана	2016-05-01 тип тарифного плана	2016-06-01 тип тарифного плана	2016-07-01 тип тарифного плана	2016-08-01 тип тарифного плана	2016-09-01 тип тарифного плана	2016-10-01 тип тарифного плана
0	3.106686e+09	KOS	2012-12-13	3491.214	1123.741	1496.170	2715.554	13083.910	11590.170	10730.143	...	пакет	пакет	пакет	пакет	пакет	пакет	пакет	пакет	пакет	пакет
1	3.108872e+09	KZT	2014-08-07	3580.384	3580.179	3562.929	3562.500	3588.527	3562.857	3580.268	...	Other	Other	Other	Other	Other	Other	Other	Other	Other	Other
2	2.820831e+09	KZT	2014-12-02	3124.260	2200.652	2183.277	2240.322	2102.366	5908.777	1836.401	...	пакет	пакет	пакет	пакет	пакет	пакет	пополнись-получи	пополнись-получи	NaN	NaN
3	2.823646e+09	KZT	2015-06-25	3116.071	3116.071	3116.071	3116.071	10276.571	3937.339	NaN	...	Other	Other	Other	Other	Other	Other	NaN	NaN	NaN	NaN
4	3.085646e+09	KZT	2015-08-19	2758.143	2546.339	996.616	1615.911	2391.982	2951.777	3208.330	...	пакет	пакет	пакет	пакет	пакет	пакет	пакет	пакет	NaN	пакет

5 rows × 113 columns

In [20]:
```
metrics = [
    'Расход абонента (тенге)',
    'Интернет трафик (Mb)',
    'исходящие международные звонки (мин)',
    'исходящие внутрисетевые платные звонки (мин)',
    'исходящие внутрисетевые бесплатные звонки (мин)',
    'исходящие внесетевые звонки (мин)',
    'входящие внутрисетевые звонки (мин)'
    'входящие внесетевые звонки (мин)',
    'расход на международные звонки (тенге)',
    'использование услуги',
    'тип тарифного плана'
]

# For each metric, melt the corresponding columns and store them in a list
melted_dataframes = []
for metric in metrics:
    # Get all columns for this particular metric
    metric_columns = [col for col in df.columns if metric in col and col.startswith('2016')]

    # Melt the DataFrame
    melted_df = pd.melt(df, id_vars=['Код абонента', 'Регион', 'Дата активации абонента'],
                        value_vars=metric_columns,
                        var_name='Date',
                        value_name=metric)

    # Convert the 'Date' column to datetime
    melted_df['Date'] = pd.to_datetime(melted_df['Date'].str.extract(r'(\d{4}-\d{2}-\d{2})')[0])

    # Append to the list
    melted_dataframes.append(melted_df)

# Combine all the melted dataframes into one
# This will align all rows by 'Код абонента', 'Регион', 'Дата активации абонента', and 'Date'
df_combined = reduce(lambda left, right: pd.merge(left, right,
                                                  on=['Код абонента', 'Регион', 'Дата активации абонента', 'Date'],
                                                  how='outer'), melted_dataframes)

df_combined
```

Out[20]:

	Код абонента	Регион	Дата активации абонента	Date	Расход абонента (тенге)	Интернет трафик (Mb)	исходящие международные звонки (мин)	исходящие внутрисетевые платные звонки (мин)	исходящие внутрисетевые бесплатные звонки (мин)	исходящие внесетевые звонки (мин)	входящие внутрисетевые звонки (мин)входящие внесетевые звонки (мин)	расход на международные звонки (тенге)	использование услуги	тип тарифного плана
0	3.106686e+09	KOS	2012-12-13	2016-01-01	3491.214	2127.105	0.0	0.250	15.033	75.850	NaN	0.000	NaN	пакет
1	3.108872e+09	KZT	2014-08-07	2016-01-01	3580.384	107436.093	0.0	0.000	0.000	0.000	NaN	0.000	NaN	Other
2	2.820831e+09	KZT	2014-12-02	2016-01-01	3124.260	13035.668	0.0	0.250	69.783	9.700	NaN	0.000	NaN	пакет
3	2.823646e+09	KZT	2015-06-25	2016-01-01	3116.071	33826.490	0.0	0.000	0.000	0.000	NaN	0.000	NaN	Other
4	3.085646e+09	KZT	2015-08-19	2016-01-01	2758.143	16963.226	0.0	0.000	26.917	68.533	NaN	0.000	NaN	пакет
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
236985	2.822292e+09	ORA	2007-01-23	2016-10-01	0.000	0.000	0.0	0.000	0.000	0.000	NaN	0.000	0.0	стандартная тарификация
236986	2.823094e+09	KZT	2015-10-26	2016-10-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
236987	2.820452e+09	PTP	2013-10-24	2016-10-01	386.956	0.000	0.1	24.833	46.433	15.517	NaN	2.679	0.0	пополнись-получи
236988	2.822534e+09	KAR	2016-01-27	2016-10-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
236989	3.085853e+09	KZT	2016-01-24	2016-10-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

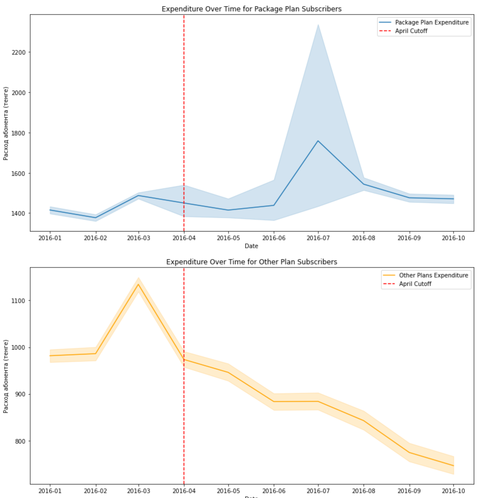
236990 rows × 14 columns

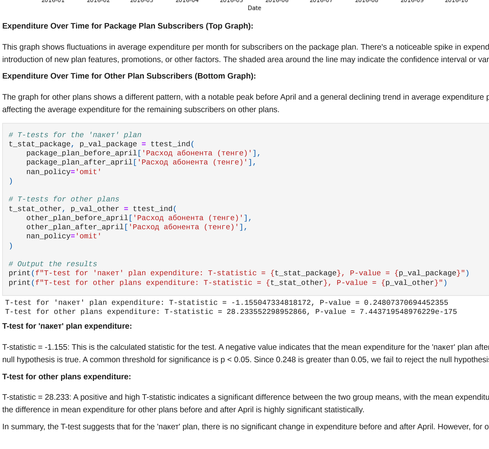
In [24]:
```
# Convert date columns to datetime
df_combined['Дата активации абонента'] = pd.to_datetime(df_combined['Дата активации абонента'])
df_combined['Date'] = pd.to_datetime(df_combined['Date'])

# Here, we'll fill NaN with 0 for simplicity
df_combined.fillna(0, inplace=True)

# Segment the data by tariff plan type
package_plan_df = df_combined[df_combined['тип тарифного плана'] == 'пакет']
other_plan_df = df_combined[df_combined['тип тарифного плана'] != 'пакет']

# Segment by time (before and after April 2016)
april_cutoff = pd.Timestamp('2016-04-01')
package_plan_before_april = package_plan_df[package_plan_df['Date'] < april_cutoff]
package_plan_after_april = package_plan_df[package_plan_df['Date'] >= april_cutoff]
other_plan_before_april = other_plan_df[other_plan_df['Date'] < april_cutoff]
other_plan_after_april = other_plan_df[other_plan_df['Date'] >= april_cutoff]

# Comparative Analysis
# Expenditure Analysis
avg_expenditure_package_before = package_plan_before_april['Расход абонента (тенге)'].mean()
avg_expenditure_package_after = package_plan_after_april['Расход абонента (тенге)'].mean()
avg_expenditure_other_before = other_plan_before_april['Расход абонента (тенге)'].mean()
avg_expenditure_other_after = other_plan_after_april['Расход абонента (тенге)'].mean()

# Usage Analysis
avg_traffic_package_before = package_plan_before_april['Интернет трафик (Mb)'].mean()
avg_traffic_package_after = package_plan_after_april['Интернет трафик (Mb)'].mean()
avg_traffic_other_before = other_plan_before_april['Интернет трафик (Mb)'].mean()
avg_traffic_other_after = other_plan_after_april['Интернет трафик (Mb)'].mean()

# Display the results
print("Average Expenditure - Package Plan Before April:", avg_expenditure_package_before)
print("Average Expenditure - Package Plan After April:", avg_expenditure_package_after)
print("Average Expenditure - Other Plan Before April:", avg_expenditure_other_before)
print("Average Expenditure - Other Plan After April:", avg_expenditure_other_after)
print("Average Internet Traffic - Package Plan Before April:", avg_traffic_package_before)
print("Average Internet Traffic - Package Plan After April:", avg_traffic_package_after)
print("Average Internet Traffic - Other Plan Before April:", avg_traffic_other_before)
print("Average Internet Traffic - Other Plan After April:", avg_traffic_other_after)
```

```
Average Expenditure - Package Plan Before April: 1428.398941538021
Average Expenditure - Package Plan After April: 1507.4306142643939
Average Expenditure - Other Plan Before April: 1031.7029886965338
Average Expenditure - Other Plan After April: 864.1830061706464
Average Internet Traffic - Package Plan Before April: 2127.4465229400885
Average Internet Traffic - Package Plan After April: 2224.0687344867
Average Internet Traffic - Other Plan Before April: 548.4536404760777
Average Internet Traffic - Other Plan After April: 248.540893864929
```

The 'naxer' plan seems to attract or cause higher spending and internet usage among its subscribers post-April. Subscribers not on the 'naxer' plan show a decrease in both expenditure and internet traffic, indicating a possible shift in the subscriber base or changes in usage patterns.

In [31]:
```
# Plot time series of expenditures for 'naxer' plan
plt.figure(figsize=(14, 7))
sns.lineplot(data=package_plan_df, x='Date', y='Расход абонента (тенге)', label='Package Plan Expenditure')
plt.axvline(pd.Timestamp('2016-04-01'), color='red', linestyle='--', label='April Cutoff')
plt.title('Expenditure Over Time for Package Plan Subscribers')
plt.legend()
plt.show()

# Plot time series of expenditures for other plans
plt.figure(figsize=(14, 7))
sns.lineplot(data=other_plan_df, x='Date', y='Расход абонента (тенге)', label='Other Plans Expenditure', color='orange')
plt.axvline(pd.Timestamp('2016-04-01'), color='red', linestyle='--', label='April Cutoff')
plt.title('Expenditure Over Time for Other Plan Subscribers')
plt.legend()
plt.show()
```





Expenditure Over Time for Package Plan Subscribers (Top Graph):

This graph shows fluctuations in average expenditure per month for subscribers on the package plan. There's a noticeable spike in expenditure after the April cutoff, suggesting an increase in spending by subscribers on the package plan during that time. This could be due to the introduction of new plan features, promotions, or other factors. The shaded area around the line may indicate the confidence interval or variance in the expenditure data, showing the range within which the true average expenditure lies.

Expenditure Over Time for Other Plan Subscribers (Bottom Graph):

The graph for other plans shows a different pattern, with a notable peak before April and a general declining trend in average expenditure post-April. This could suggest that subscribers on other plans either reduced their spending after April or possibly migrated to the package plan, affecting the average expenditure for the remaining subscribers on other plans.

In [39]:
```
# T-tests for the 'naxer' plan
t_stat_package, p_val_package = ttest_ind(
    package_plan_before_april['Расход абонента (тенге)'],
    package_plan_after_april['Расход абонента (тенге)'],
    nan_policy='omit'
)

# T-tests for other plans
t_stat_other, p_val_other = ttest_ind(
    other_plan_before_april['Расход абонента (тенге)'],
    other_plan_after_april['Расход абонента (тенге)'],
    nan_policy='omit'
)

# Output the results
print(f"T-test for 'naxer' plan expenditure: T-statistic = {t_stat_package}, P-value = {p_val_package}")
print(f"T-test for other plans expenditure: T-statistic = {t_stat_other}, P-value = {p_val_other}")
```

```
T-test for 'naxer' plan expenditure: T-statistic = -1.1550473348818172, P-value = 0.2480737069445355
T-test for other plans expenditure: T-statistic = 28.233552298952866, P-value = 7.443719548976229e-175
```

T-test for 'naxer' plan expenditure:

T-statistic = -1.155: This is the calculated statistic for the test. A negative value indicates that the mean expenditure for the 'naxer' plan after April is lower than before April, but it's not very far from zero. P-value = 0.248: The P-value tells us the probability of observing the data assuming the null hypothesis is true. A common threshold for significance is p < 0.05. Since 0.248 is greater than 0.05, we fail to reject the null hypothesis. This means there is not enough evidence to say there is a significant difference in expenditure for the 'naxer' plan before and after April.

T-test for other plans expenditure:

T-statistic = 28.233: A positive and high T-statistic indicates a significant difference between the two group means, with the mean expenditure for other plans after April being higher than before April. P-value ≈ 0: The P-value is practically zero, which is much less than 0.05, indicating that the difference in mean expenditure for other plans before and after April is highly significant statistically.

In summary, the T-test suggests that for the 'naxer' plan, there is no significant change in expenditure before and after April. However, for other plans, there is a significant change, with expenditures increasing after April.