In [2]:
```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import scipy.stats as stats

from scipy.stats import skew, kurtosis

import seaborn as sns

import statistics as stat

#Loading the CSV of the default dataset
df = pd.read_csv(r'C:\Users\mmorg\Desktop\D207 Assessment Files\medical_clean.csv')

#Turn categorical values into quantitative data
df['Marital_numeric'] = df['Marital']
dict_marital = {"Marital_numeric": {"Never Married": 0,"Separated": 1,"Widowed": 2,"Divorced": 3
df.replace(dict_marital, inplace=True)

df['Gender_numeric'] = df['Gender']
dict_gender = {"Gender_numeric": {"Prefer not to answer": 0,"Male": 1,"Female": 2}}
df.replace(dict_gender, inplace=True)

df['ReAdmis_numeric'] = df['ReAdmis']
dict_ReAdmis = {"ReAdmis_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_ReAdmis, inplace=True)

df['Soft_drink_numeric'] = df['Soft_drink']
dict_Soft_drink = {"Soft_drink_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_Soft_drink, inplace=True)

df['Initial_admin_numeric'] = df['Initial_admin']
dict_Initial_admin = {"Initial_admin_numeric": {"Emergency Admission": 0, "Elective Admission":
df.replace(dict_Initial_admin, inplace=True)

df['HighBlood_numeric'] = df['HighBlood']
dict_HighBlood = {"HighBlood_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_HighBlood, inplace=True)

df['Stroke_numeric'] = df['Stroke']
dict_stroke = {"Stroke_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_stroke, inplace=True)

df['Complication_risk_numeric'] = df['Complication_risk']
dict_complication = {"Complication_risk_numeric": {"Low": 0, "Medium": 1, "High": 2}}
df.replace(dict_complication, inplace=True)

df['Arthritis_numeric'] = df['Arthritis']
dict_arthritis = {"Arthritis_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_arthritis, inplace=True)

df['Diabetes_numeric'] = df['Diabetes']
dict_diabetes = {"Diabetes_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_diabetes, inplace=True)

df['Hyperlipidemia_numeric'] = df['Hyperlipidemia']
dict_hyperlipidemia = {"Hyperlipidemia_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_hyperlipidemia, inplace=True)

df['BackPain_numeric'] = df['BackPain']
dict_backpain = {"BackPain_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_backpain, inplace=True)

df['Allergic_rhinitis_numeric'] = df['Allergic_rhinitis']
```

```
dict_allergies = {"Allergic_rhinitis_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_allergies, inplace=True)

df['Reflux_esophagitis_numeric'] = df['Reflux_esophagitis']
dict_reflux = {"Reflux_esophagitis_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_reflux, inplace=True)

df['Asthma_numeric'] = df['Asthma']
dict_asthma = {"Asthma_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_asthma, inplace=True)

df['Services_numeric'] = df['Services']
dict_services = {"Services_numeric": {"Blood Work": 0, "Intravenous": 1, "CT Scan": 2, "MRI": 3
df.replace(dict_services, inplace=True)

##Univariate Stats Dataframe
def unistats(df):
    output_df = pd.DataFrame(columns=['Count', 'Missing', 'Unique', 'Dtype', 'Numeric', 'Mean',

    for col in df:
        if pd.api.types.is_numeric_dtype(df[col]):
            output_df.loc[col] = [df[col].count(), df[col].isnull().sum(), df[col].nunique(), d
        else:
            output_df.loc[col] = [df[col].count(), df[col].isnull().sum(), df[col].nunique(), d
    return output_df.sort_values(by=['Numeric', 'Skew', 'Unique'], ascending=False)

df.drop(columns=['CaseOrder', 'Customer_id', 'Interaction', 'UID', 'City', 'State', 'Lat', 'Lng
print(unistats(df))
```

	Count	Missing	Unique	Dtype	Numeric
vitD_supp	10000	0	6	int64	True
Stroke_numeric	10000	0	2	int64	True
Children	10000	0	11	int64	True
Income	10000	0	9993	float64	True
Soft_drink_numeric	10000	0	2	int64	True
Services_numeric	10000	0	4	int64	True
Diabetes_numeric	10000	0	2	int64	True
Full_meals_eaten	10000	0	8	int64	True
Asthma_numeric	10000	0	2	int64	True
Additional_charges	10000	0	9418	float64	True
Hyperlipidemia_numeric	10000	0	2	int64	True
Arthritis_numeric	10000	0	2	int64	True
ReAdmis_numeric	10000	0	2	int64	True
Initial_admin_numeric	10000	0	3	int64	True
Allergic_rhinitis_numeric	10000	0	2	int64	True
HighBlood_numeric	10000	0	2	int64	True
BackPain_numeric	10000	0	2	int64	True
Reflux_esophagitis_numeric	10000	0	2	int64	True
Initial_days	10000	0	9997	float64	True
TotalCharge	10000	0	9997	float64	True
VitD_levels	10000	0	9976	float64	True
Age	10000	0	72	int64	True
Marital_numeric	10000	0	5	int64	True
Doc_visits	10000	0	9	int64	True
Complication_risk_numeric	10000	0	3	int64	True
Marital	10000	0	5	object	False
Services	10000	0	4	object	False
Gender	10000	0	3	object	False
Initial_admin	10000	0	3	object	False
Complication_risk	10000	0	3	object	False
Gender_numeric	10000	0	3	object	False
ReAdmis	10000	0	2	object	False
Soft_drink	10000	0	2	object	False
HighBlood	10000	0	2	object	False
Stroke	10000	0	2	object	False
Overweight	10000	0	2	object	False
Arthritis	10000	0	2	object	False
Diabetes	10000	0	2	object	False
Hyperlipidemia	10000	0	2	object	False
BackPain	10000	0	2	object	False
Anxiety	10000	0	2	object	False
Allergic_rhinitis	10000	0	2	object	False
Reflux_esophagitis	10000	0	2	object	False
Asthma	10000	0	2	object	False

	Mean	Mode	Min
vitD_supp	0.3989	0	0
Stroke_numeric	0.1993	0	0
Children	2.0972	0.0	0.0
Income	40490.49516	14572.4	154.08
Soft_drink_numeric	0.2575	0	0
Services_numeric	0.672	0	0
Diabetes_numeric	0.2738	0	0
Full_meals_eaten	1.0014	0	0
Asthma_numeric	0.2893	0	0
Additional_charges	12934.528587	3883.66416	3125.703
Hyperlipidemia_numeric	0.3372	0	0
Arthritis_numeric	0.3574	0	0
ReAdmis_numeric	0.3669	0	0
Initial_admin_numeric	0.7376	0	0
Allergic_rhinitis_numeric	0.3941	0	0
HighBlood_numeric	0.409	0	0
BackPain_numeric	0.4114	0	0
Reflux_esophagitis_numeric	0.4135	0	0
Initial_days	34.455299	63.54432	1.001981
TotalCharge	5312.172769	7555.452	1938.312067

VitD_levels	17.964262	15.26009	9.806483
Age	53.5117	47.0	18.0
Marital_numeric	2.0052	2	0
Doc_visits	5.0122	5	1
Complication_risk_numeric	1.1233	1	0
Marital	-	-	-
Services	-	-	-
Gender	-	-	-
Initial_admin	-	-	-
Complication_risk	-	-	-
Gender_numeric	-	-	-
ReAdmis	-	-	-
Soft_drink	-	-	-
HighBlood	-	-	-
Stroke	-	-	-
Overweight	-	-	-
Arthritis	-	-	-
Diabetes	-	-	-
Hyperlipidemia	-	-	-
BackPain	-	-	-
Anxiety	-	-	-
Allergic_rhinitis	-	-	-
Reflux_esophagitis	-	-	-
Asthma	-	-	-

	25%	Median	75% Quartile	Max	\
vitD_supp	0.0	0.0	1.0	5	
Stroke_numeric	0.0	0.0	0.0	1	
Children	0.0	1.0	3.0	10.0	
Income	19598.775	33768.42	54296.4025	207249.1	
Soft_drink_numeric	0.0	0.0	1.0	1	
Services_numeric	0.0	0.0	1.0	3	
Diabetes_numeric	0.0	0.0	1.0	1	
Full_meals_eaten	0.0	1.0	2.0	7	
Asthma_numeric	0.0	0.0	1.0	1	
Additional_charges	7986.487755	11573.977735	15626.49	30566.07	
Hyperlipidemia_numeric	0.0	0.0	1.0	1	
Arthritis_numeric	0.0	0.0	1.0	1	
ReAdmis_numeric	0.0	0.0	1.0	1	
Initial_admin_numeric	0.0	0.0	1.0	2	
Allergic_rhinitis_numeric	0.0	0.0	1.0	1	
HighBlood_numeric	0.0	0.0	1.0	1	
BackPain_numeric	0.0	0.0	1.0	1	
Reflux_esophagitis_numeric	0.0	0.0	1.0	1	
Initial_days	7.896215	35.836244	61.16102	71.98149	
TotalCharge	3179.374015	5213.952	7459.69975	9180.728	
VitD_levels	16.626439	17.951122	19.347963	26.394449	
Age	36.0	53.0	71.0	89.0	
Marital_numeric	1.0	2.0	3.0	4	
Doc_visits	4.0	5.0	6.0	9	
Complication_risk_numeric	1.0	1.0	2.0	2	
Marital	-	-	-	-	
Services	-	-	-	-	
Gender	-	-	-	-	
Initial_admin	-	-	-	-	
Complication_risk	-	-	-	-	
Gender_numeric	-	-	-	-	
ReAdmis	-	-	-	-	
Soft_drink	-	-	-	-	
HighBlood	-	-	-	-	
Stroke	-	-	-	-	
Overweight	-	-	-	-	
Arthritis	-	-	-	-	
Diabetes	-	-	-	-	
Hyperlipidemia	-	-	-	-	
BackPain	-	-	-	-	
Anxiety	-	-	-	-	

Allergic_rhinitis	-	-	-	-
Reflux_esophagitis	-	-	-	-
Asthma	-	-	-	-

	Std	Skew	Kurt
vitD_supp	0.628505	1.550205	2.330763
Stroke_numeric	0.399494	1.505705	0.267202
Children	2.163659	1.448013	2.076321
Income	28521.153293	1.405899	2.74569
Soft_drink_numeric	0.437279	1.109354	-0.769488
Services_numeric	0.832758	1.069764	0.345281
Diabetes_numeric	0.44593	1.014712	-0.970553
Full_meals_eaten	1.008117	1.009461	1.042727
Asthma_numeric	0.45346	0.929485	-1.136285
Additional_charges	6542.601544	0.831842	-0.142684
Hyperlipidemia_numeric	0.472777	0.688834	-1.525813
Arthritis_numeric	0.479258	0.595206	-1.646059
ReAdmis_numeric	0.481983	0.552412	-1.69518
Initial_admin_numeric	0.825115	0.51916	-1.339272
Allergic_rhinitis_numeric	0.488681	0.433498	-1.812442
HighBlood_numeric	0.491674	0.370238	-1.863296
BackPain_numeric	0.492112	0.360153	-1.870664
Reflux_esophagitis_numeric	0.492486	0.35135	-1.876929
Initial_days	26.309341	0.070286	-1.754525
TotalCharge	2180.393838	0.069661	-1.668267
VitD_levels	2.017231	0.032435	-0.022112
Age	20.638538	0.005117	-1.189527
Marital_numeric	1.413426	-0.000908	-1.294478
Doc_visits	1.045734	-0.018563	0.025999
Complication_risk_numeric	0.730172	-0.194687	-1.111062
Marital	-	-	-
Services	-	-	-
Gender	-	-	-
Initial_admin	-	-	-
Complication_risk	-	-	-
Gender_numeric	-	-	-
ReAdmis	-	-	-
Soft_drink	-	-	-
HighBlood	-	-	-
Stroke	-	-	-
Overweight	-	-	-
Arthritis	-	-	-
Diabetes	-	-	-
Hyperlipidemia	-	-	-
BackPain	-	-	-
Anxiety	-	-	-
Allergic_rhinitis	-	-	-
Reflux_esophagitis	-	-	-
Asthma	-	-	-

In [4]:
```
# Bivariate: Numeric to numeric: Correlation
# Bivariate: Numeric to categorical: one-way ANOVA (3+ groups) or t-test (2 groups)
# Bivariate: categorical to categorical: Chi-square

def bivstats(df, label):
   from scipy import stats
   import pandas as pd

   #Create an empty dataframe to store output
   output_df = pd.DataFrame(columns=['r', 'p-value'])

   for col in df:
       if pd.api.types.is_numeric_dtype(df[col]): #Only calculate r, p-value, for numeric columr
           r, p = stats.pearsonr(df[label], df[col])
           output_df.loc[col] = [round(r, 3), round(p, 3)]

   return output_df.sort_values(by=['p-value'], ascending=True)

bivstats(df, 'ReAdmis_numeric')
```

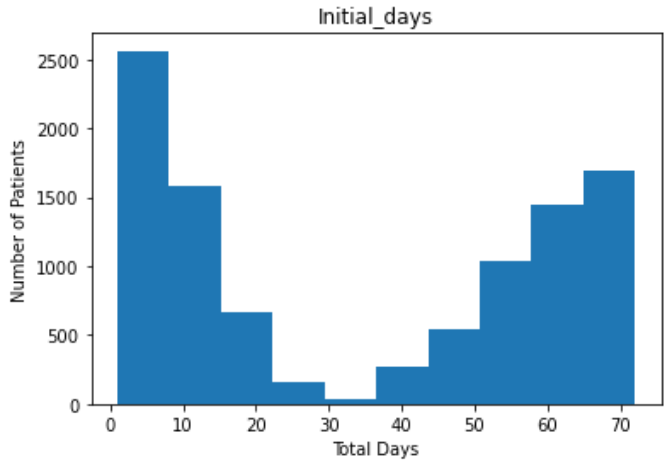
Out[4]:

	r	p-value
ReAdmis_numeric	1.000	0.000
TotalCharge	0.844	0.000
Initial_days	0.851	0.000
Children	0.024	0.019
Initial_admin_numeric	-0.018	0.069
Asthma_numeric	-0.017	0.087
Age	0.016	0.114
Services_numeric	0.014	0.152
Additional_charges	0.014	0.173
BackPain_numeric	0.013	0.183
Full_meals_eaten	-0.012	0.224
Income	-0.012	0.250
vitD_supp	0.011	0.270
Soft_drink_numeric	0.008	0.441
Arthritis_numeric	0.008	0.444
Marital_numeric	-0.006	0.547
Reflux_esophagitis_numeric	0.005	0.588
Allergic_rhinitis_numeric	-0.005	0.642
Hyperlipidemia_numeric	0.004	0.667
VitD_levels	0.004	0.683
Complication_risk_numeric	-0.003	0.746
Diabetes_numeric	-0.003	0.760
HighBlood_numeric	0.002	0.820
Stroke_numeric	0.001	0.927
Doc_visits	0.000	0.980

In [5]:
```
##Selected Initial_days for t-test due to low p-value
stats.ttest_ind(df['ReAdmis_numeric'], df['Initial_days'])
```

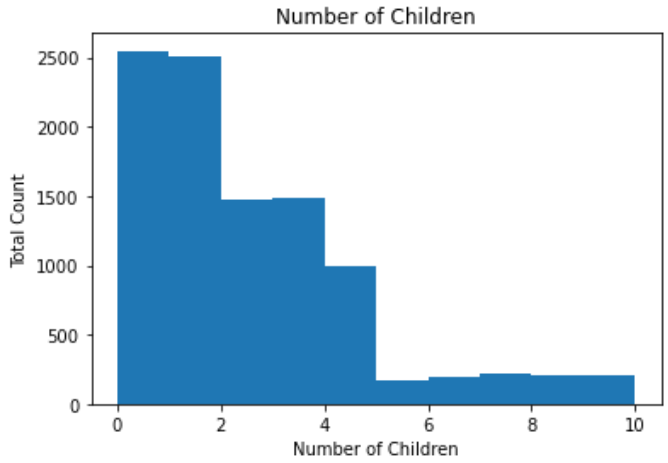
Out[5]:
```
Ttest_indResult(statistic=-129.54592813419822, pvalue=0.0)
```

In [6]:
```
##Initial_days distribution
plt.hist(df.Initial_days)
plt.xlabel('Total Days')
plt.ylabel('Number of Patients')
plt.title('Initial_days')
plt.show()

##Initial_days univariate statistics
print('Initial Days Stats')
print('min:', df.Initial_days.min())
print('25th Quantile:', df.Initial_days.quantile(.25))
print('50th Quantile:', df.Initial_days.quantile(.50))
print('75th Quantile:', df.Initial_days.quantile(.75))
print('max:', df.Initial_days.max())
print('mean:', df.Initial_days.mean())
print('median:', df.Initial_days.median())
print('mode:', df.Initial_days.mode().values[0])
print('Std:', df.Initial_days.std())
print('skew:', skew(df.Initial_days, bias=False))
print('kurtosis:', kurtosis(df.Initial_days, bias=False))
```



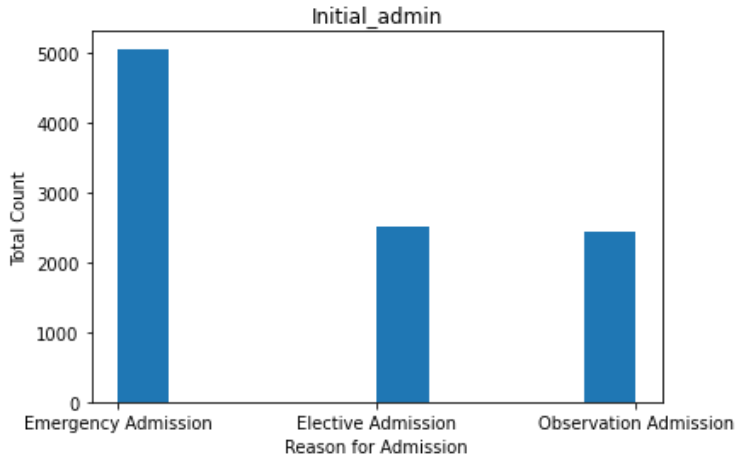

```
Initial Days Stats
min: 1.001980919
25th Quantile: 7.896214698
50th Quantile: 35.83624435
75th Quantile: 61.16102
max: 71.98149
mean: 34.45529926595239
median: 35.83624435
mode: 63.54432
Std: 26.30934131161786
skew: 0.07028608266045329
kurtosis: -1.7545246170896873
```

In [7]:
```
##Children distribution
plt.hist(df.Children)
plt.xlabel('Number of Children')
plt.ylabel('Total Count')
plt.title('Number of Children')
plt.show()

##Children univariate statistics
print('Children Stats')
print('min:', df.Children.min())
print('25th Quantile:', df.Children.quantile(.25))
print('50th Quantile:', df.Children.quantile(.50))
print('75th Quantile:', df.Children.quantile(.75))
print('max:', df.Children.max())
print('mean:', df.Children.mean())
print('median:', df.Children.median())
print('mode:', df.Children.mode().values[0])
print('Std:', df.Children.std())
print('skew:', skew(df.Children, bias=False))
print('kurtosis:', kurtosis(df.Children, bias=False))
```



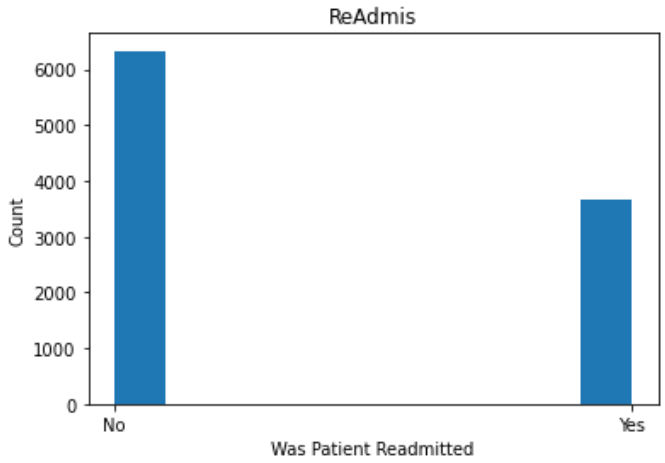

```
Children Stats
min: 0
25th Quantile: 0.0
50th Quantile: 1.0
75th Quantile: 3.0
max: 10
mean: 2.0972
median: 1.0
mode: 0
Std: 2.16365900779899
skew: 1.4480126219332756
kurtosis: 2.076321273332364
```

In [8]:
```
##Initial_admin distribution
plt.hist(df.Initial_admin)
plt.xlabel('Reason for Admission')
plt.ylabel('Total Count')
plt.title('Initial_admin')
plt.show()

##Initial_admin_numeric univariate statistics
print('Initial_admin_numeric Stats')
print('min:', df.Initial_admin_numeric.min())
print('25th Quantile:', df.Initial_admin_numeric.quantile(.25))
print('50th Quantile:', df.Initial_admin_numeric.quantile(.50))
print('75th Quantile:', df.Initial_admin_numeric.quantile(.75))
print('max:', df.Initial_admin_numeric.max())
print('mean:', df.Initial_admin_numeric.mean())
print('median:', df.Initial_admin_numeric.median())
print('mode:', df.Initial_admin_numeric.mode().values[0])
print('Std:', df.Initial_admin_numeric.std())
print('skew:', skew(df.Initial_admin_numeric, bias=False))
print('kurtosis:', kurtosis(df.Initial_admin_numeric, bias=False))
```




```
Initial_admin_numeric Stats
min: 0
25th Quantile: 0.0
50th Quantile: 0.0
75th Quantile: 1.0
max: 2
mean: 0.7376
median: 0.0
mode: 0
Std: 0.8251147322840162
skew: 0.5191601076816872
kurtosis: -1.3392723170631167
```

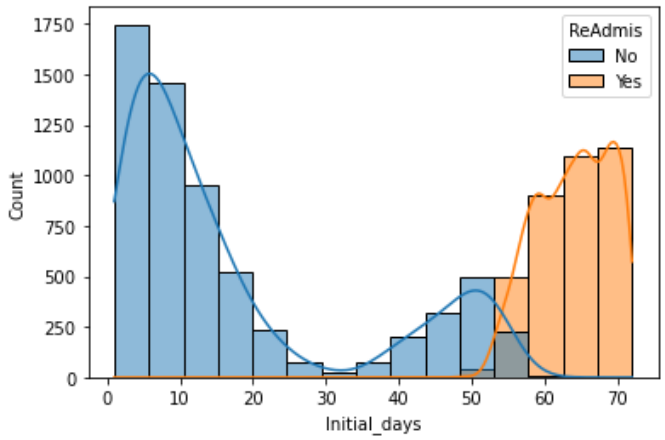
In [9]:
```
##ReAdmis_numeric distribution
plt.hist(df.ReAdmis)
plt.xlabel('Was Patient Readmitted')
plt.ylabel('Count')
plt.title('ReAdmis')
plt.show()

##ReAdmis_numeric univariate statistics
print('ReAdmis_numeric Stats')
print('min:', df.ReAdmis_numeric.min())
print('25th Quantile:', df.ReAdmis_numeric.quantile(.25))
print('50th Quantile:', df.ReAdmis_numeric.quantile(.50))
print('75th Quantile:', df.ReAdmis_numeric.quantile(.75))
print('max:', df.ReAdmis_numeric.max())
print('mean:', df.ReAdmis_numeric.mean())
print('median:', df.ReAdmis_numeric.median())
print('mode:', df.ReAdmis_numeric.mode().values[0])
print('Std:', df.ReAdmis_numeric.std())
print('skew:', skew(df.ReAdmis_numeric, bias=False))
print('kurtosis:', kurtosis(df.ReAdmis_numeric, bias=False))
```




```
ReAdmis_numeric Stats
min: 0
25th Quantile: 0.0
50th Quantile: 0.0
75th Quantile: 1.0
max: 1
mean: 0.3669
median: 0.0
mode: 0
Std: 0.48198300878982964
skew: 0.5524121095443897
kurtosis: -1.695179937226946
```

In [10]:
```
sns.histplot(data=df, x="Initial_days", hue="ReAdmis", kde=True)
```

Out[10]: <AxesSubplot:xlabel='Initial_days', ylabel='Count'>

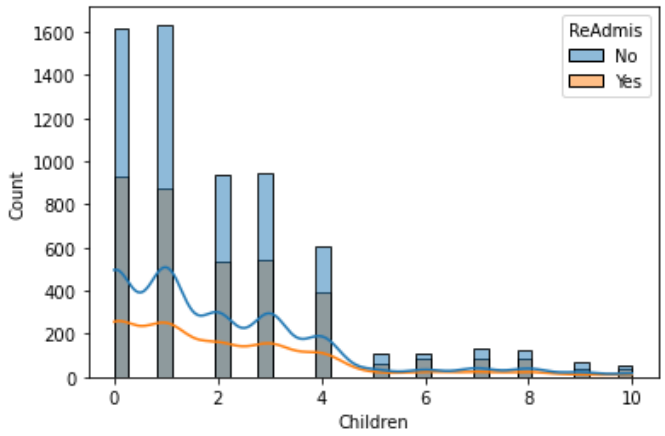


In [11]:
```
cor = df['Initial_days'].corr(df['ReAdmis_numeric'])
print(cor)
```

```
0.8508616016470936
```

In [12]:
```
sns.histplot(data=df, x="Children", hue="ReAdmis", kde=True)
```

Out[12]: <AxesSubplot:xlabel='Children', ylabel='Count'>



In [13]:
```
cor = df['Children'].corr(df['ReAdmis_numeric'])
print(cor)
```

```
0.0235315217234477
```