In [80]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [81]:
```python
df = pd.read_csv("WA_Fn-UseC_-Telco-Customer-Churn.csv")
```

In [82]:
```python
df.head()
```

Out[82]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...

5 rows × 21 columns

In [83]:
```python
df = df.dropna()
```

In [84]:
```python
df.isnull().sum()
```

Out[84]:
```
customerID          0
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges        0
Churn               0
dtype: int64
```

In [85]:
```python
df.columns
```

Out[85]:
```
Index(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport',
       'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling',
       'PaymentMethod', 'MonthlyCharges', 'TotalCharges', 'Churn'],
      dtype='object')
```

In [86]:
```python
df.describe()
```

Out[86]:

	SeniorCitizen	tenure	MonthlyCharges
count	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692
std	0.368612	24.559481	30.090047
min	0.000000	0.000000	18.250000
25%	0.000000	9.000000	35.500000
50%	0.000000	29.000000	70.350000
75%	0.000000	55.000000	89.850000
max	1.000000	72.000000	118.750000

1.)Convert categorical features to dummy variables (Yes/No counts as categorical

In [87]:
```python
df.drop('customerID', axis=1, inplace=True)
```

In [88]:
```python
from sklearn import preprocessing
from sklearn.preprocessing import LabelEncoder

for i in df.columns:
    if 'object' in str(df[str(i)].dtype):
        df[str(i)] = df[str(i)].astype('category').cat.codes

print(df.shape)
df.head()
```

(7043, 20)

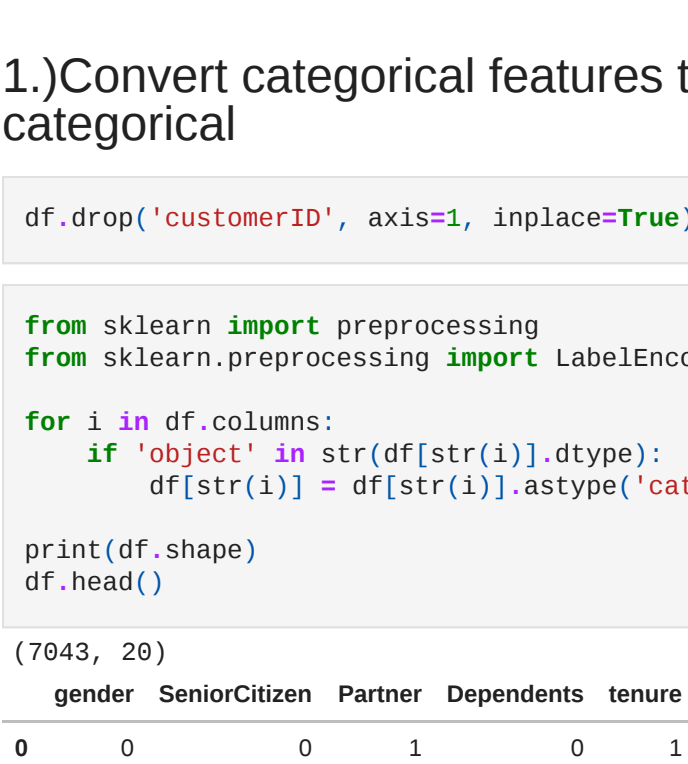
Out[88]:

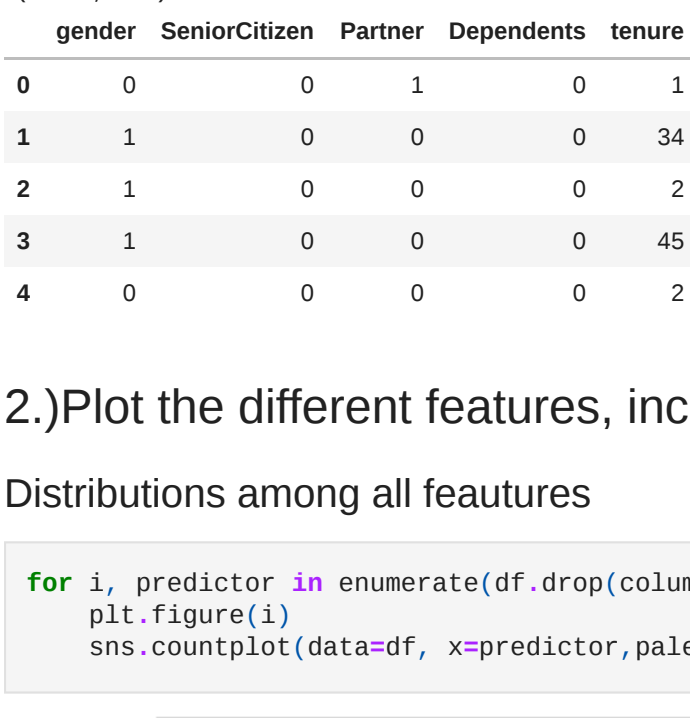
	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	De
0	0	0	1	0	1	0	1	0	0	2	
1	1	0	0	0	34	1	0	0	2	0	
2	1	0	0	0	2	1	0	0	2	2	
3	1	0	0	0	45	0	1	0	2	0	
4	0	0	0	0	2	1	0	1	0	0	

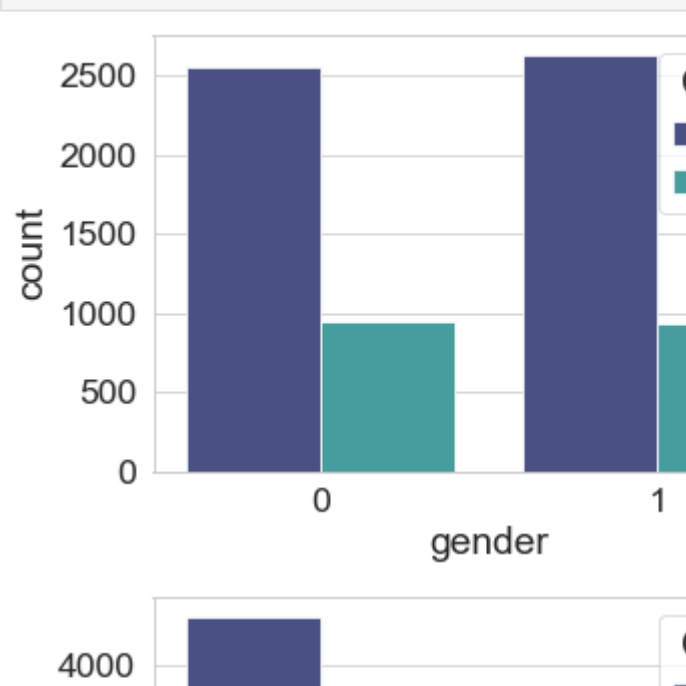
2.)Plot the different features, including the distribution of the target variable

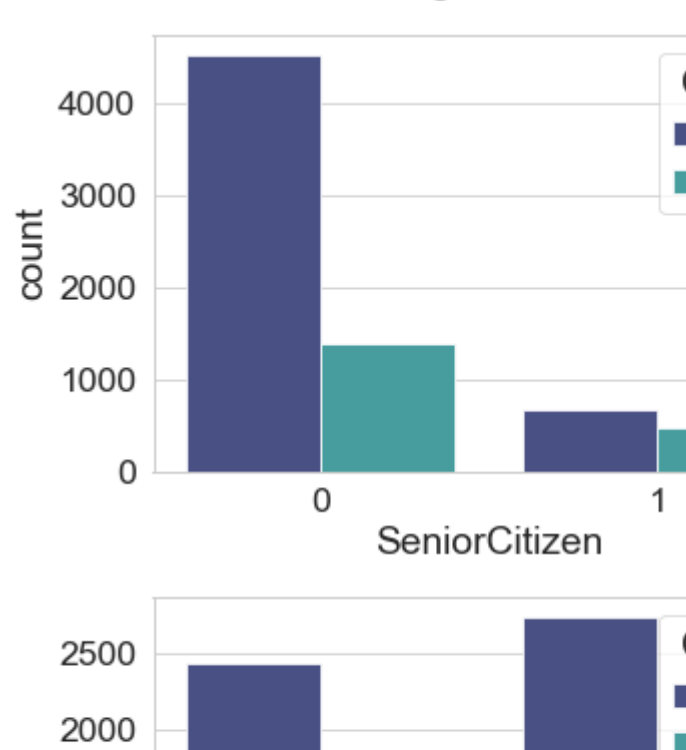
Distributions among all feautures

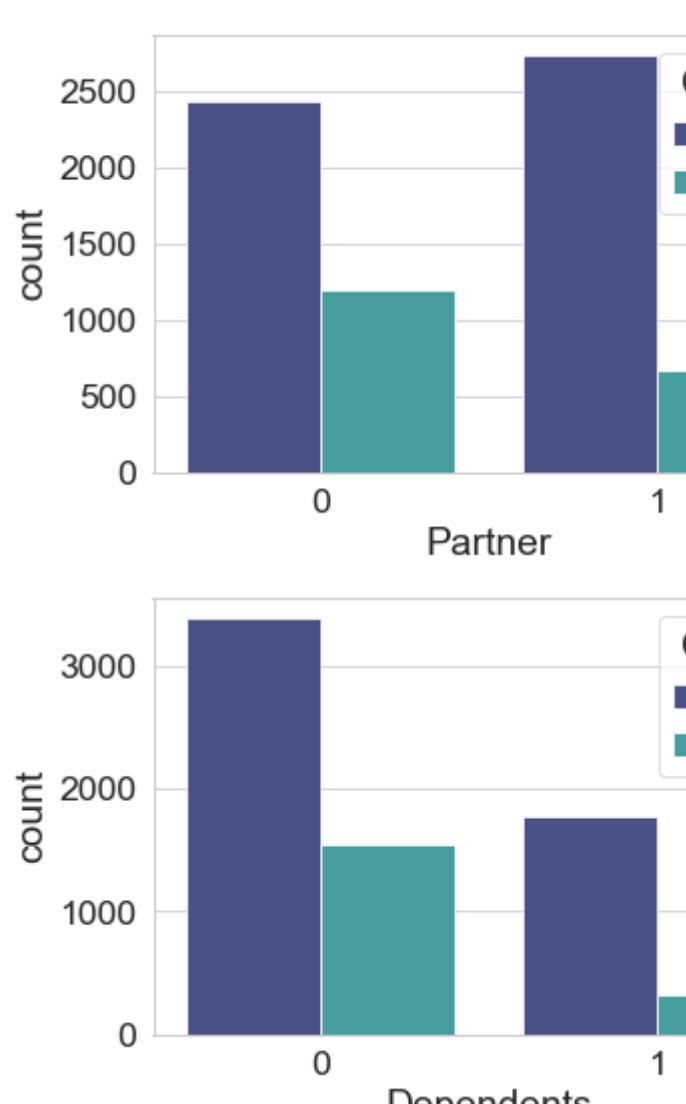
In [89]:
```python
for i, predictor in enumerate(df.drop(columns=['Churn', 'TotalCharges', 'MonthlyCharges'])):
    plt.figure(i)
    sns.countplot(data=df, x=predictor,palette='mako', hue='Churn')
```

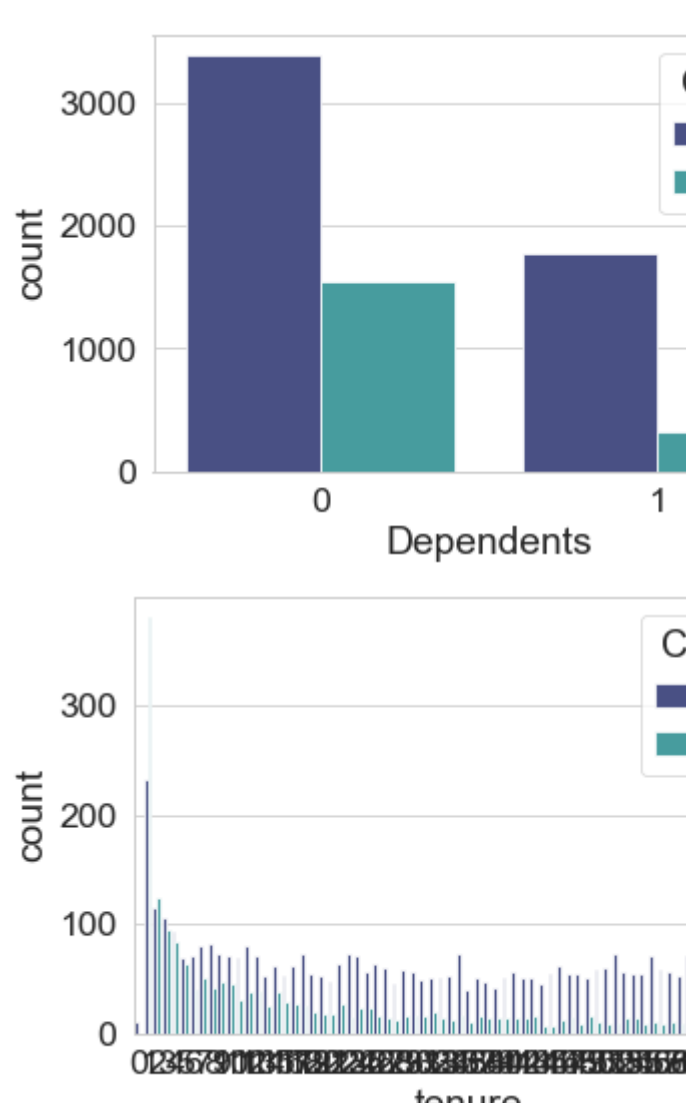


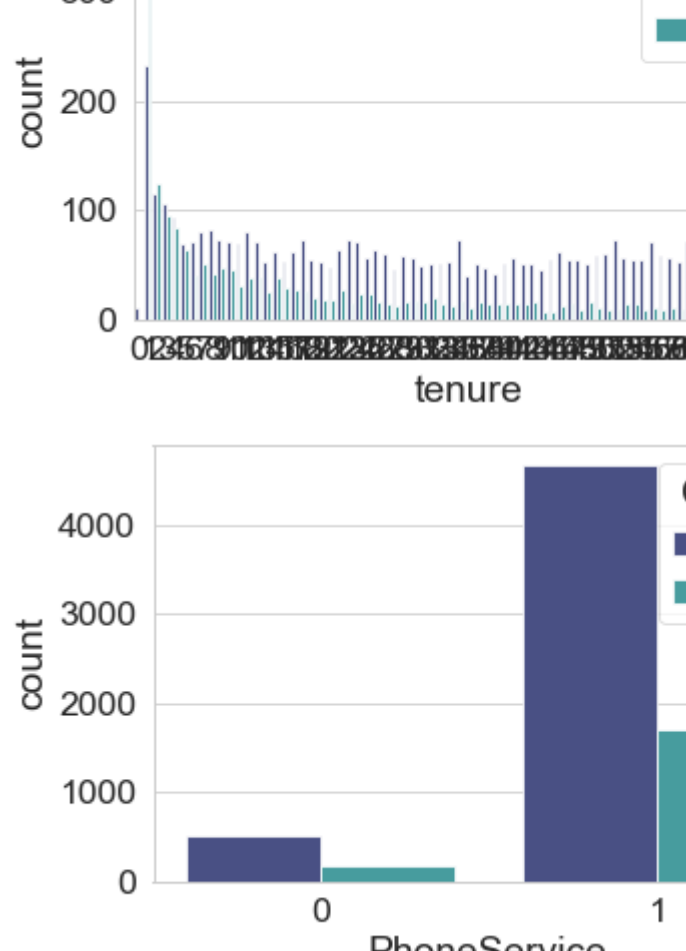


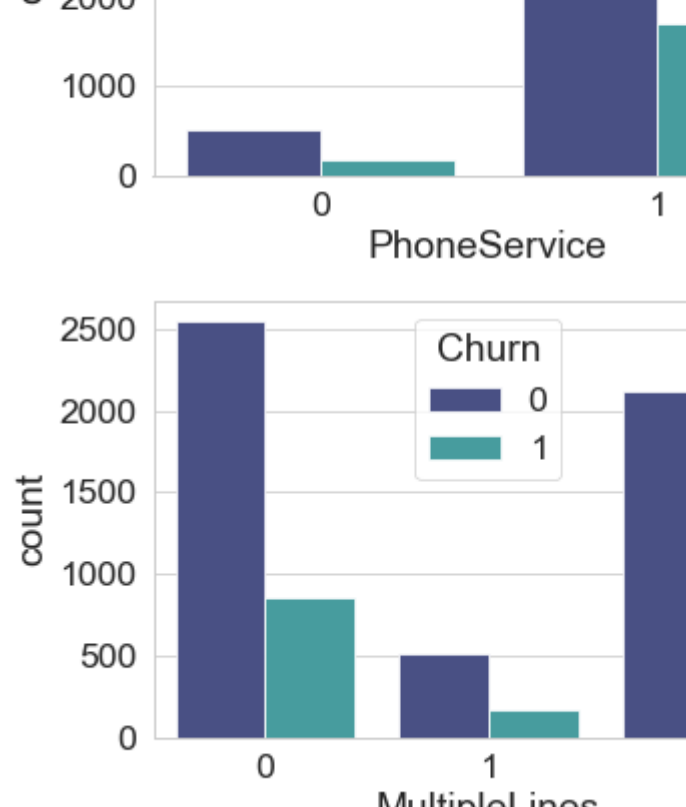


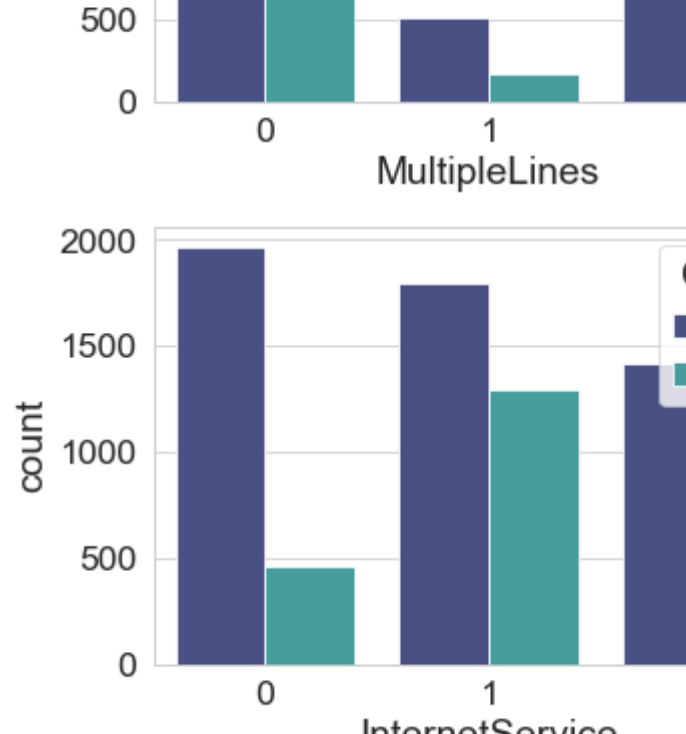


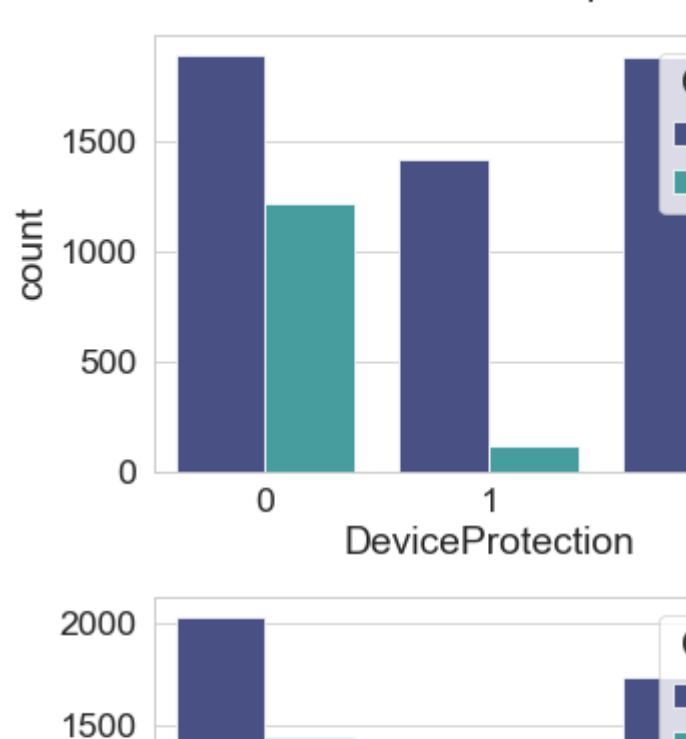


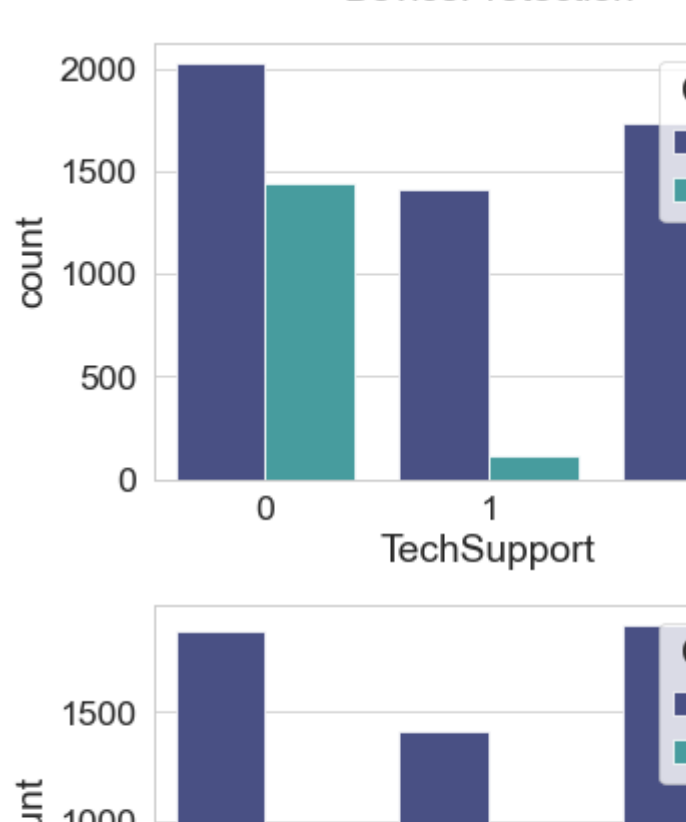


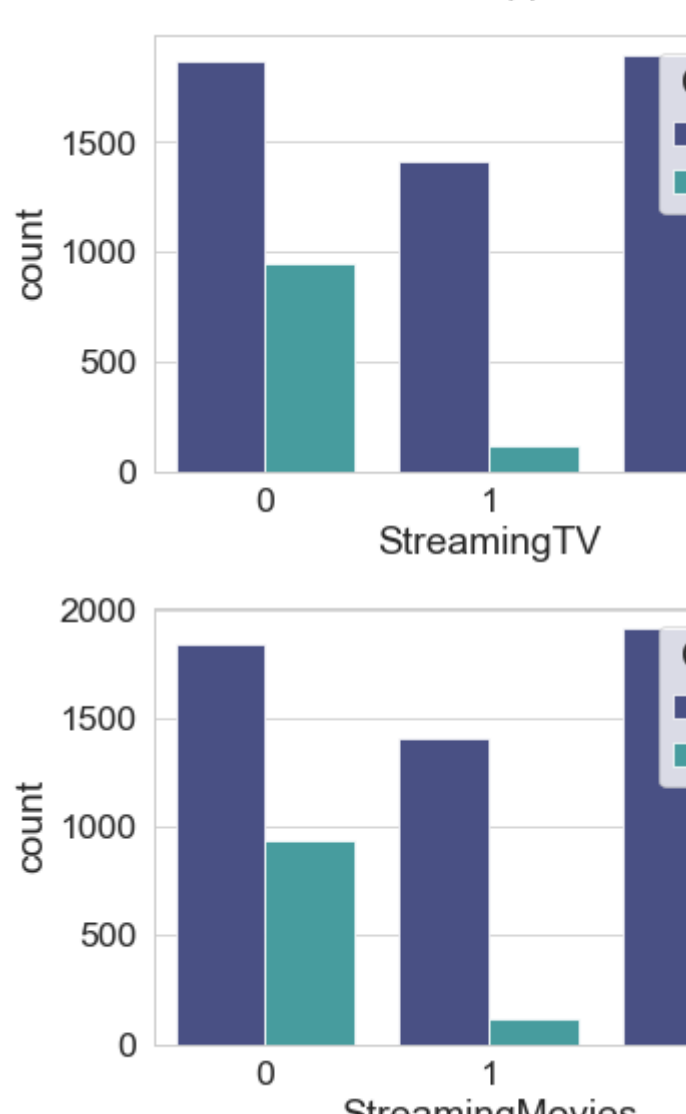


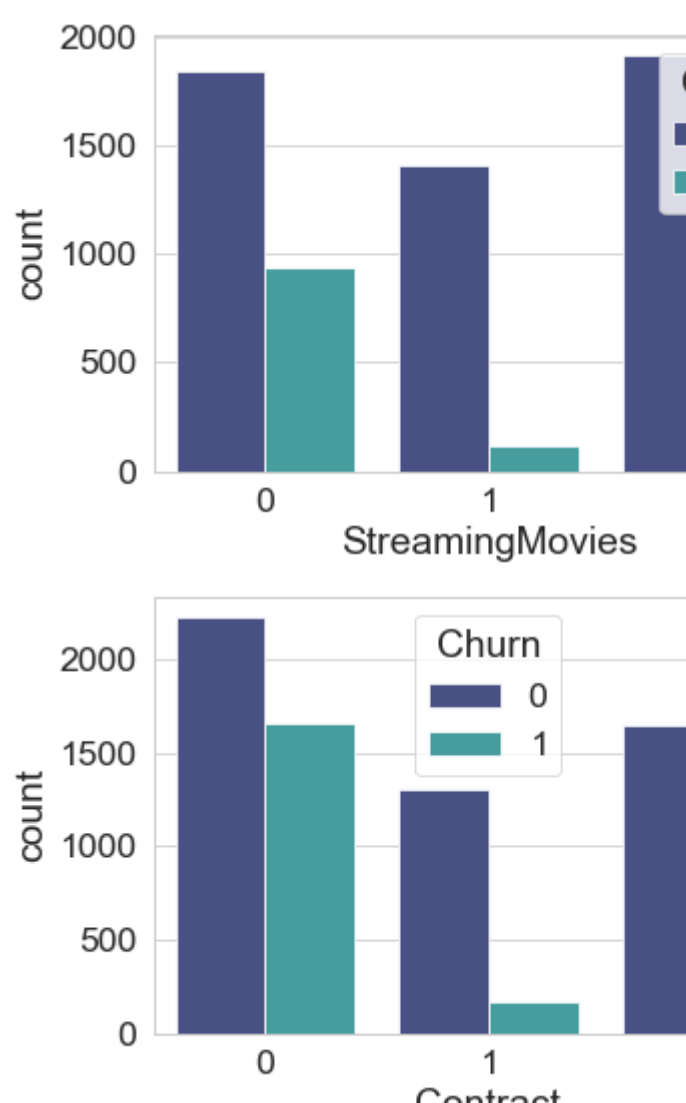


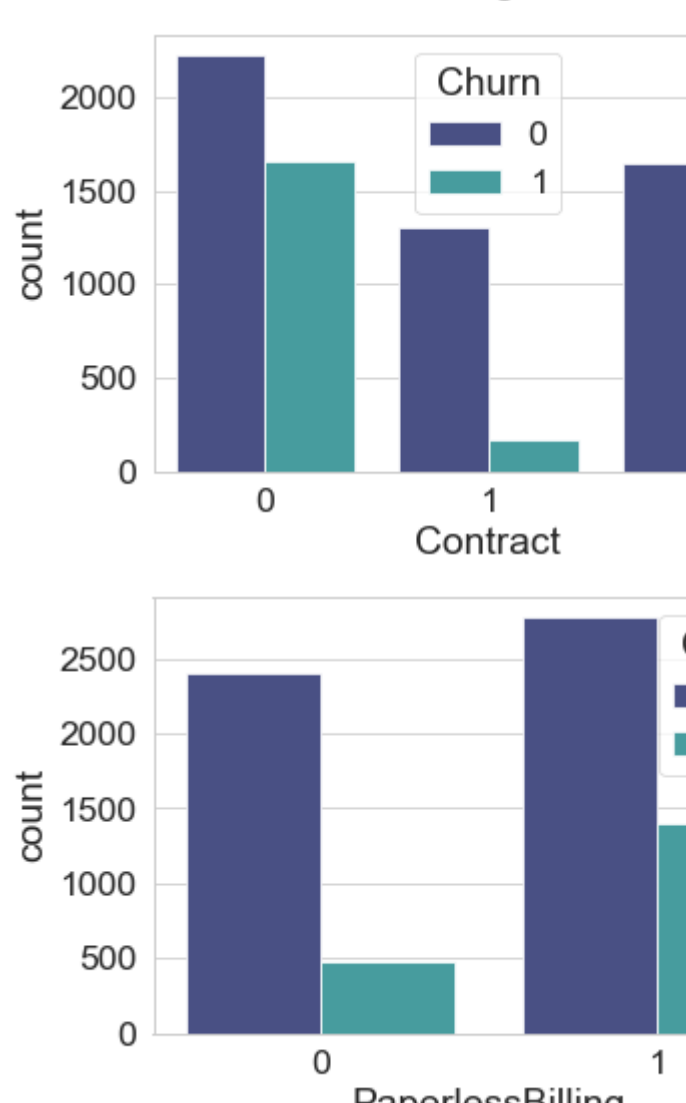


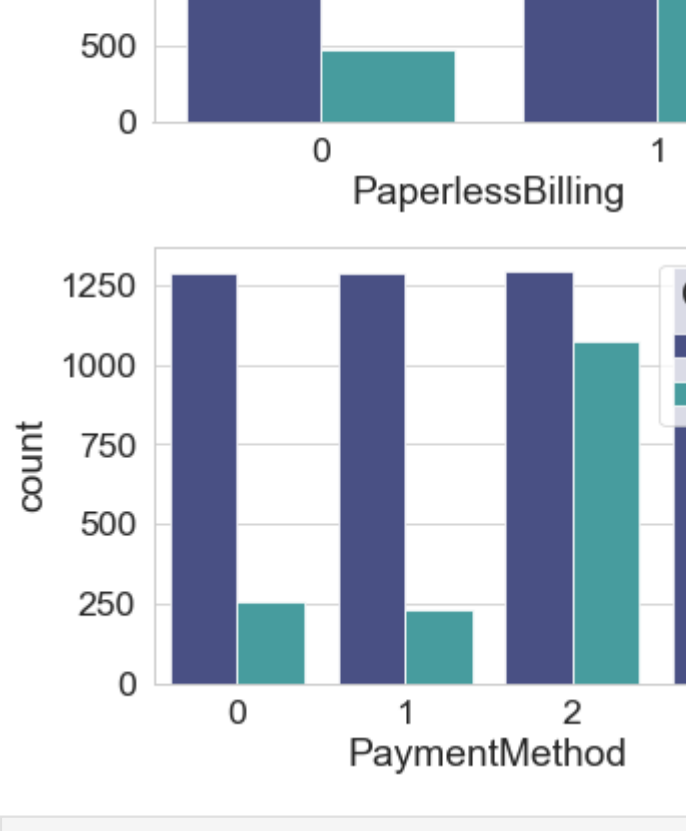






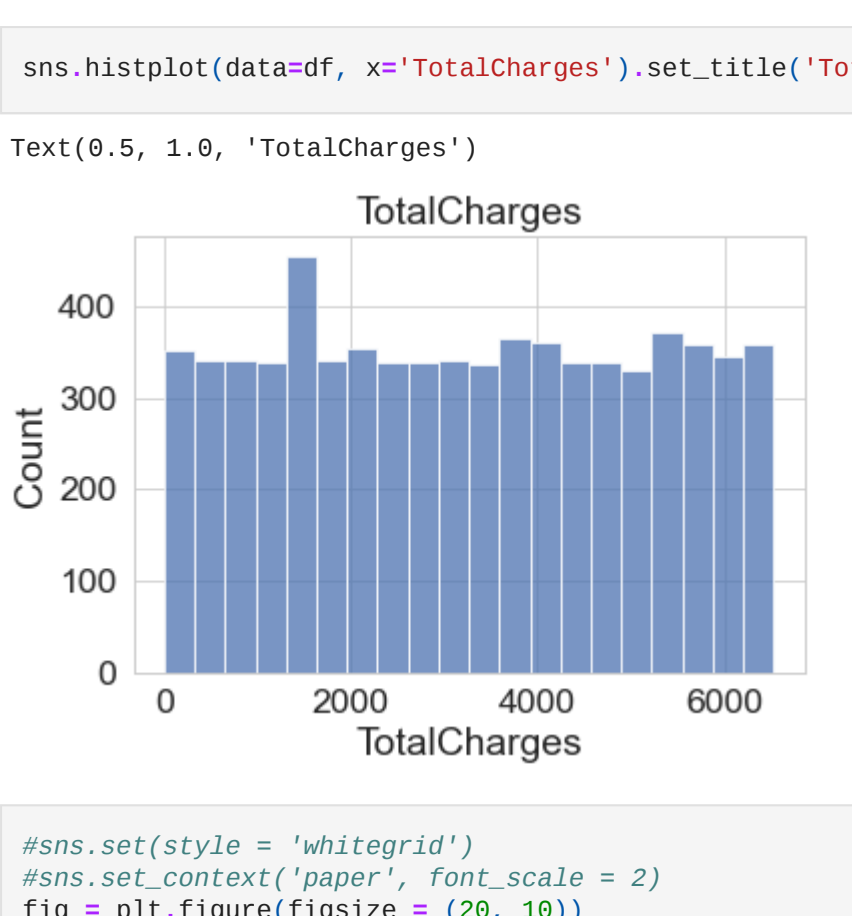






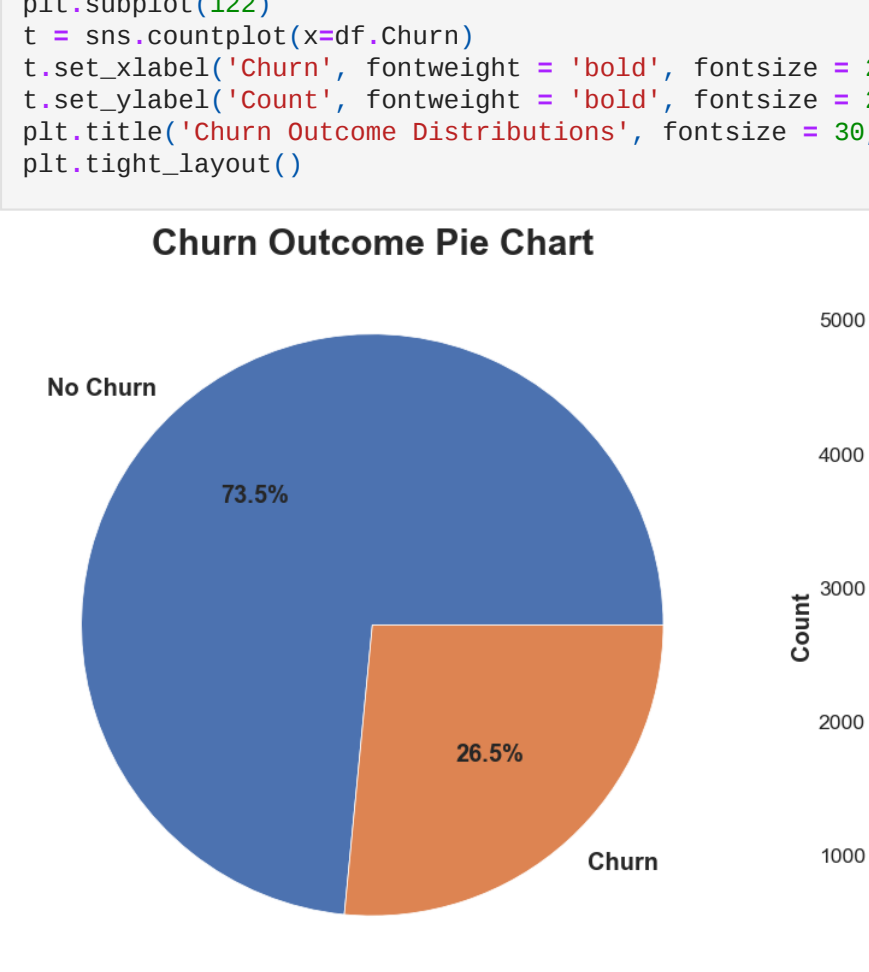
In [90]:
```python
sns.histplot(data=df, x='MonthlyCharges').set_title('MonthlyCharges')
```

Out[90]: Text(0.5, 1.0, 'MonthlyCharges')

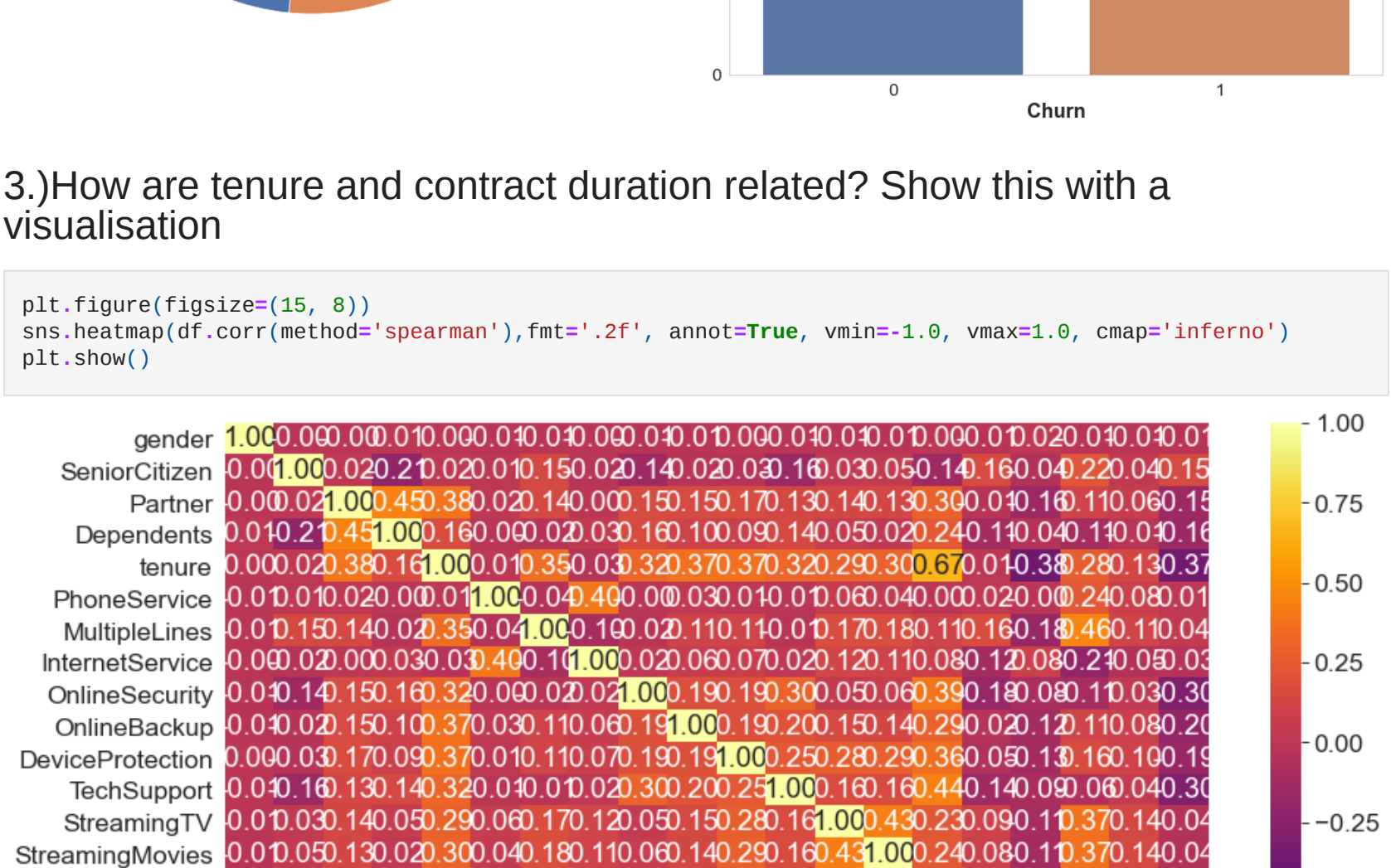


In [91]:
```python
sns.histplot(data=df, x='TotalCharges').set_title('TotalCharges')
```

Out[91]: Text(0.5, 1.0, 'TotalCharges')

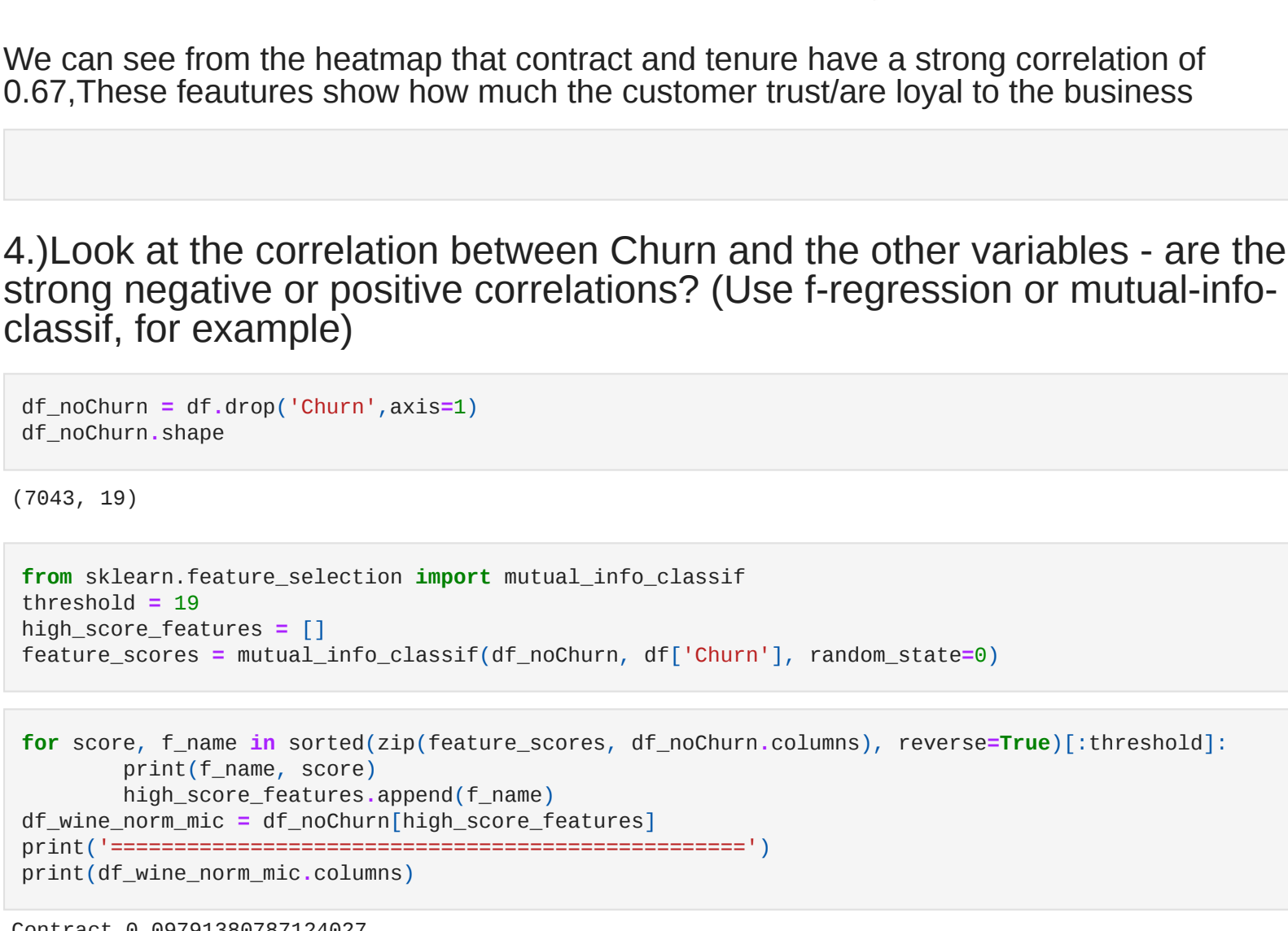


In [92]:
```python
#sns.set(style = 'whitegrid')
#sns.set_context('paper', font_scale = 2)
fig = plt.figure(figsize = (20, 10))
plt.subplot(121)
plt.pie(df.Churn.value_counts(),labels = ['No Churn', 'Churn'], autopct = '%.1f%%', radius = 1, textprops=
plt.title('Churn Outcome Pie Chart', fontsize = 30, fontweight = 'bold')
plt.subplot(122)
t = sns.countplot(x=df.Churn)
t.set_xlabel('Churn', fontweight = 'bold', fontsize = 20)
t.set_ylabel('Count', fontweight = 'bold', fontsize = 20)
plt.title('Churn Outcome Distributions', fontsize = 30, fontweight = 'bold')
plt.tight_layout()
```



3.)How are tenure and contract duration related? Show this with a visualisation

In [93]:
```python
plt.figure(figsize=(15, 8))
sns.heatmap(df.corr(method='spearman'),fmt='.2f', annot=True, vmin=-1.0, vmax=1.0, cmap='inferno')
plt.show()
```



We can see from the heatmap that contract and tenure have a strong correlation of 0.67,These feautures show how much the customer trust/are loyal to the business

In [ ]:

4.)Look at the correlation between Churn and the other variables - are there strong negative or positive correlations? (Use f-regression or mutual-info-classif, for example)

In [94]:
```python
df_noChurn = df.drop('Churn',axis=1)
df_noChurn.shape
```

Out[94]: (7043, 19)

In [95]:
```python
from sklearn.feature_selection import mutual_info_classif
threshold = 19
high_score_features = []
feature_scores = mutual_info_classif(df_noChurn, df['Churn'], random_state=0)
```

In [96]:
```python
for score, f_name in sorted(zip(feature_scores, df_noChurn.columns), reverse=True)[:threshold]:
        print(f_name, score)
        high_score_features.append(f_name)
df_wine_norm_mic = df_noChurn[high_score_features]
print('==================================================')
print(df_wine_norm_mic.columns)
```

```
Contract 0.09791380787124027
tenure 0.07852220063906934
OnlineSecurity 0.06603907832105582
InternetService 0.06493301826603037
OnlineBackup 0.05648588278698741
TechSupport 0.05643734021727176
PaymentMethod 0.04972324111948501
MonthlyCharges 0.04556515319247634
DeviceProtection 0.03856150242023859
StreamingMovies 0.03667907806258919
TotalCharges 0.034282788820399146
StreamingTV 0.027777026747217937
PaperlessBilling 0.023418550976705532
SeniorCitizen 0.015732399035717215
Dependents 0.014170438590713985
Partner 0.010140701466378577
gender 0.0
PhoneService 0.0
MultipleLines 0.0
==================================================
Index(['Contract', 'tenure', 'OnlineSecurity', 'InternetService',
       'OnlineBackup', 'TechSupport', 'PaymentMethod', 'MonthlyCharges',
       'DeviceProtection', 'StreamingMovies', 'TotalCharges', 'StreamingTV',
       'PaperlessBilling', 'SeniorCitizen', 'Dependents', 'Partner', 'gender',
       'PhoneService', 'MultipleLines'],
      dtype='object')
```

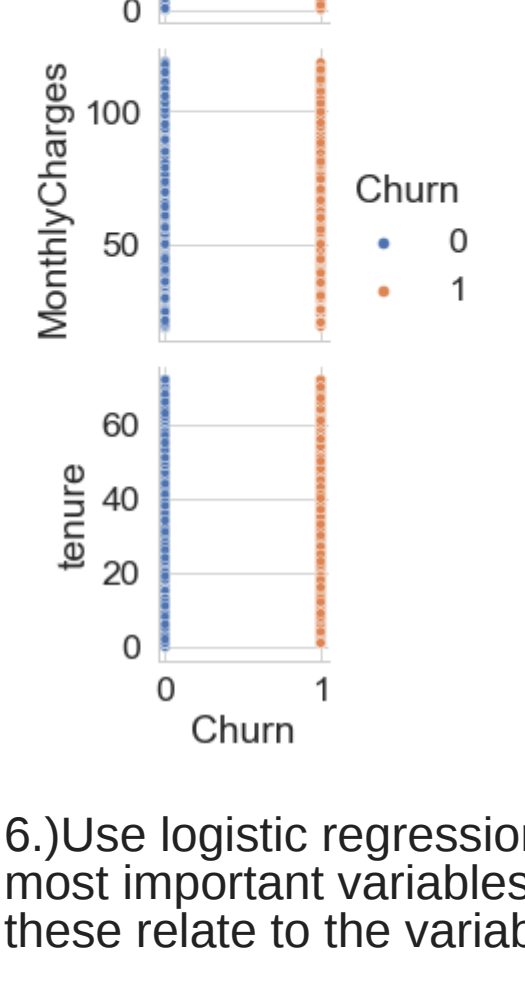
The mutual_info_classif meauseres the dependancy of all the feautures against Churn column, the higher the value the more dependant they are to each other

So from the results we can see the top 5 feautures that have a strong correlation with churn are: Contract, tenure, onlineSecurity, InternetService and OnlineBackup

5.)Produce plots to look at churn vs tenure, contract, age, monthly and total charges

In [112…
```python
sns.pairplot(data=df,
            x_vars=['Churn'],
            y_vars=["TotalCharges", "MonthlyCharges", "tenure"],
            hue='Churn')
```

Out[112… <seaborn.axisgrid.PairGrid at 0x1fb2ad91580>



6.)Use logistic regression (l1 norm) and a random forest to get a list of the most important variables.How different are they from each other, and how do these relate to the variables from the correlations above?

L1 Norm

In [116…
```python
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    df.drop(labels=['Churn'], axis=1),
    df['Churn'],
    test_size=0.3,
    random_state=0)
X_train.shape, X_test.shape
```

Out[116… ((4930, 19), (2113, 19))

In [129…
```python
from sklearn.linear_model import Lasso, LogisticRegression
from sklearn.feature_selection import SelectFromModel

sel_ = SelectFromModel(LogisticRegression(C=0.01, penalty='l1',solver='liblinear'))
sel_.fit((X_train.fillna(0)), y_train)
```

Out[129…
```
SelectFromModel(estimator=LogisticRegression(C=0.01, penalty='l1',
                                             solver='liblinear'))
```

In [130…
```python
sel_.get_support()
```

Out[130…
```
array([False, False, False, False,  True,  True,  True, False,  True,
        True, False,  True, False, False,  True, False,  True,  True,
        True])
```

In [132…
```python
selected_feat = X_train.columns[(sel_.get_support())]
print('total features: {}'.format((X_train.shape[1])))
print('selected features: {}'.format(len(selected_feat)))
print('features with coefficients shrank to zero: {}'.format(
      np.sum(sel_.estimator_.coef_ == 0)))
```

```
total features: 19
selected features: 10
features with coefficients shrank to zero: 9
```

In [133…
```python
selected_feat
```

Out[133…
```
Index(['tenure', 'PhoneService', 'MultipleLines', 'OnlineSecurity',
       'OnlineBackup', 'TechSupport', 'Contract', 'PaymentMethod',
       'MonthlyCharges', 'TotalCharges'],
      dtype='object')
```

Random Forest

In [134…
```python
from sklearn.ensemble import RandomForestRegressor

sel2_ = SelectFromModel(RandomForestRegressor(random_state=0))
sel2_.fit((X_train.fillna(0)), y_train)
```

Out[134… SelectFromModel(estimator=RandomForestRegressor(random_state=0))

In [135…
```python
sel2_.get_support()
```

Out[135…
```
array([False, False, False, False,  True, False, False, False, False,
       False, False, False, False, False,  True, False, False,  True,
        True])
```

In [141…
```python
selected_feat_rf = X_train.columns[(sel2_.get_support())]
print('total features: {}'.format((X_train.shape[1])))
print('selected features: {}'.format(len(selected_feat_rf)))
print('features with coefficients shrank to zero: {}'.format(
      np.sum(sel_.estimator_.coef_ == 0)))
```

```
total features: 19
selected features: 4
features with coefficients shrank to zero: 9
```

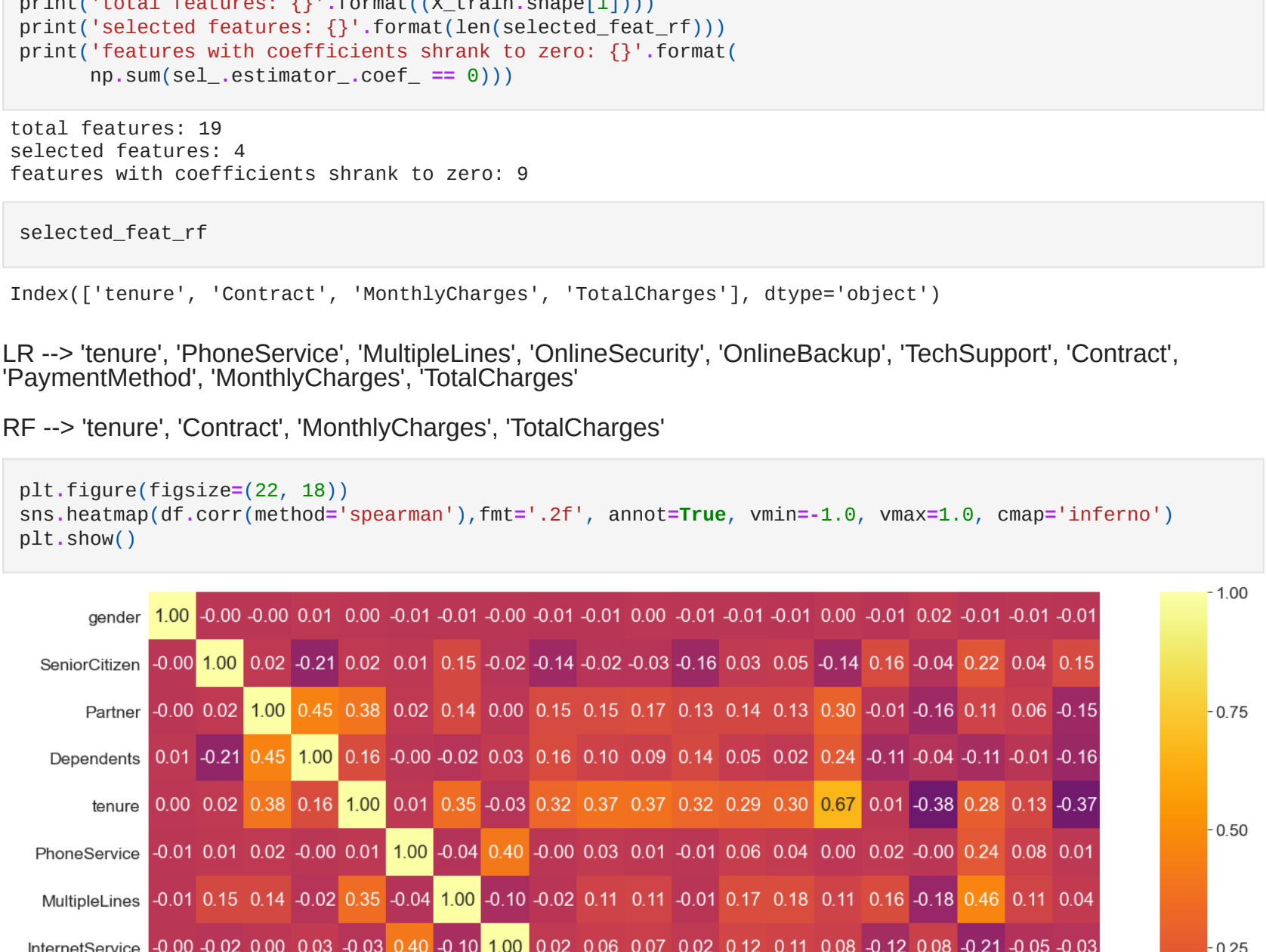
In [143…
```python
selected_feat_rf
```

Out[143… Index(['tenure', 'Contract', 'MonthlyCharges', 'TotalCharges'], dtype='object')

LR --> 'tenure', 'PhoneService', 'MultipleLines', 'OnlineSecurity', 'OnlineBackup', 'TechSupport', 'Contract', 'PaymentMethod', 'MonthlyCharges', 'TotalCharges'

RF --> 'tenure', 'Contract', 'MonthlyCharges', 'TotalCharges'

In [138…
```python
plt.figure(figsize=(22, 18))
sns.heatmap(df.corr(method='spearman'),fmt='.2f', annot=True, vmin=-1.0, vmax=1.0, cmap='inferno')
plt.show()
```



from the above results we can conclude that RF has less feautures selected than Logistic Regression It has the similar feautures except for Contract.Both contain positive and negative correlations