```
In [36]: import platform
         import pandas as pd
         import sklearn
         import numpy as np
         import graphviz
         import matplotlib
         import matplotlib.pyplot as plt

         %matplotlib inline
```

```
In [37]: df = pd.read_csv('Churn.csv')
         df.shape

Out[37]: (7043, 21)
```

```
In [38]: df = df.dropna(how="all")  # remove samples with all missing values
         df.shape

Out[38]: (7043, 21)
```

```
In [39]: df = df[~df.duplicated()] # remove duplicates
         df.shape

Out[39]: (7043, 21)
```

```
In [40]: total_charges_filter = df.TotalCharges == " "
         df = df[~total_charges_filter]
         df.shape

Out[40]: (7032, 21)
```

```
In [41]: df.TotalCharges = pd.to_numeric(df.TotalCharges)
```

```
In [42]: df.describe(include='all')
```

Out[42]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneServic
count	7032	7032	7032.000000	7032	7032	7032.000000	7032
unique	7032	2	NaN	2	2	NaN	2
top	2351-RRBUE	Male	NaN	No	No	NaN	Yes
freq	1	3549	NaN	3639	4933	NaN	6352
mean	NaN	NaN	0.162400	NaN	NaN	32.421786	NaN
std	NaN	NaN	0.368844	NaN	NaN	24.545260	NaN
min	NaN	NaN	0.000000	NaN	NaN	1.000000	NaN
25%	NaN	NaN	0.000000	NaN	NaN	9.000000	NaN
50%	NaN	NaN	0.000000	NaN	NaN	29.000000	NaN
75%	NaN	NaN	0.000000	NaN	NaN	55.000000	NaN
max	NaN	NaN	1.000000	NaN	NaN	72.000000	NaN

11 rows × 21 columns

In [43]:

```
categorical_features = [
    "gender",
    "SeniorCitizen",
    "Partner",
    "Dependents",
    "PhoneService",
    "MultipleLines",
    "InternetService",
    "OnlineSecurity",
    "OnlineBackup",
    "DeviceProtection",
    "TechSupport",
    "StreamingTV",
    "StreamingMovies",
    "Contract",
    "PaperlessBilling",
    "PaymentMethod",
]
numerical_features = ["tenure", "MonthlyCharges", "TotalCharges"]
target = "Churn"
```

In [44]:

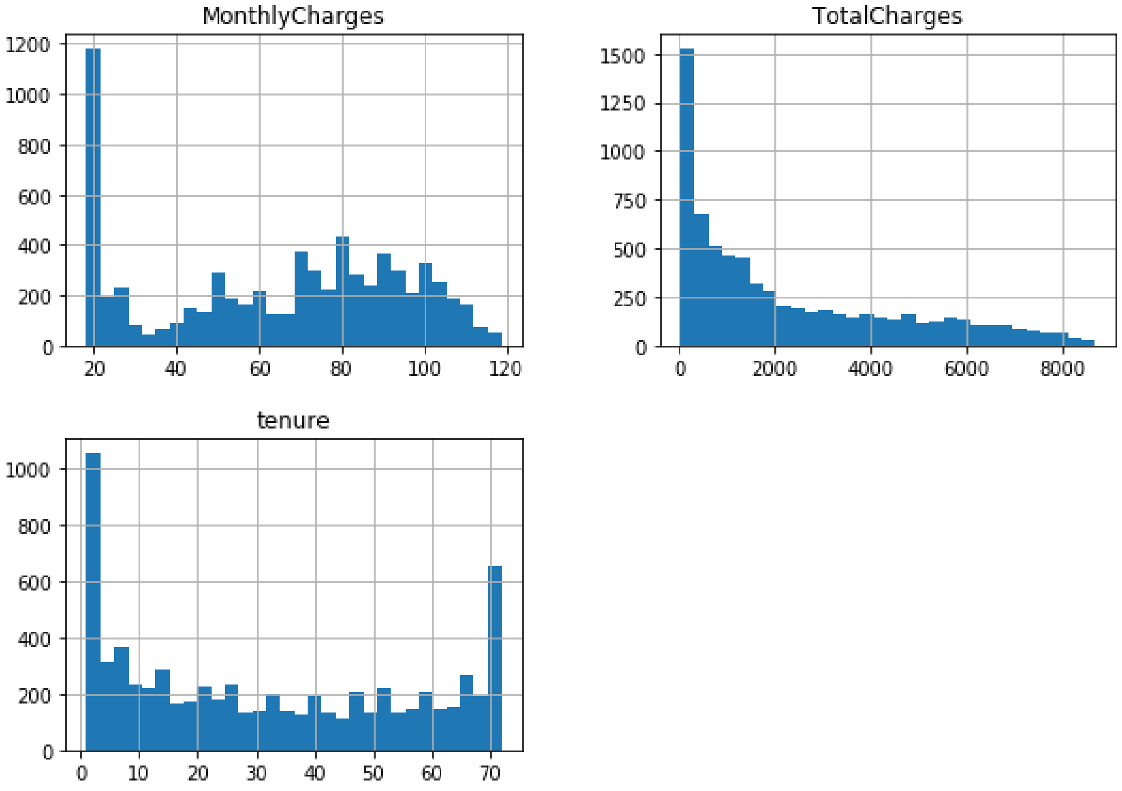
```
df[numerical_features].describe()
```

Out[44]:

	tenure	MonthlyCharges	TotalCharges
count	7032.000000	7032.000000	7032.000000
mean	32.421786	64.798208	2283.300441
std	24.545260	30.085974	2266.771362
min	1.000000	18.250000	18.800000
25%	9.000000	35.587500	401.450000
50%	29.000000	70.350000	1397.475000
75%	55.000000	89.862500	3794.737500
max	72.000000	118.750000	8684.800000

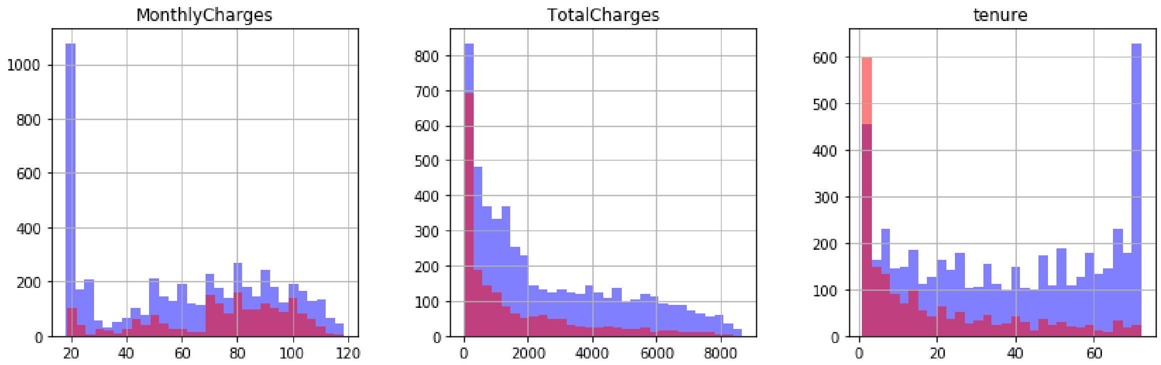
```
In [45]: df[numerical_features].hist(bins=30, figsize=(10, 7))
```

```
Out[45]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x00000259EF9312B0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000259F0BE8908>],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x00000259F0BB6978>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000259F21866D8>]],
               dtype=object)
```



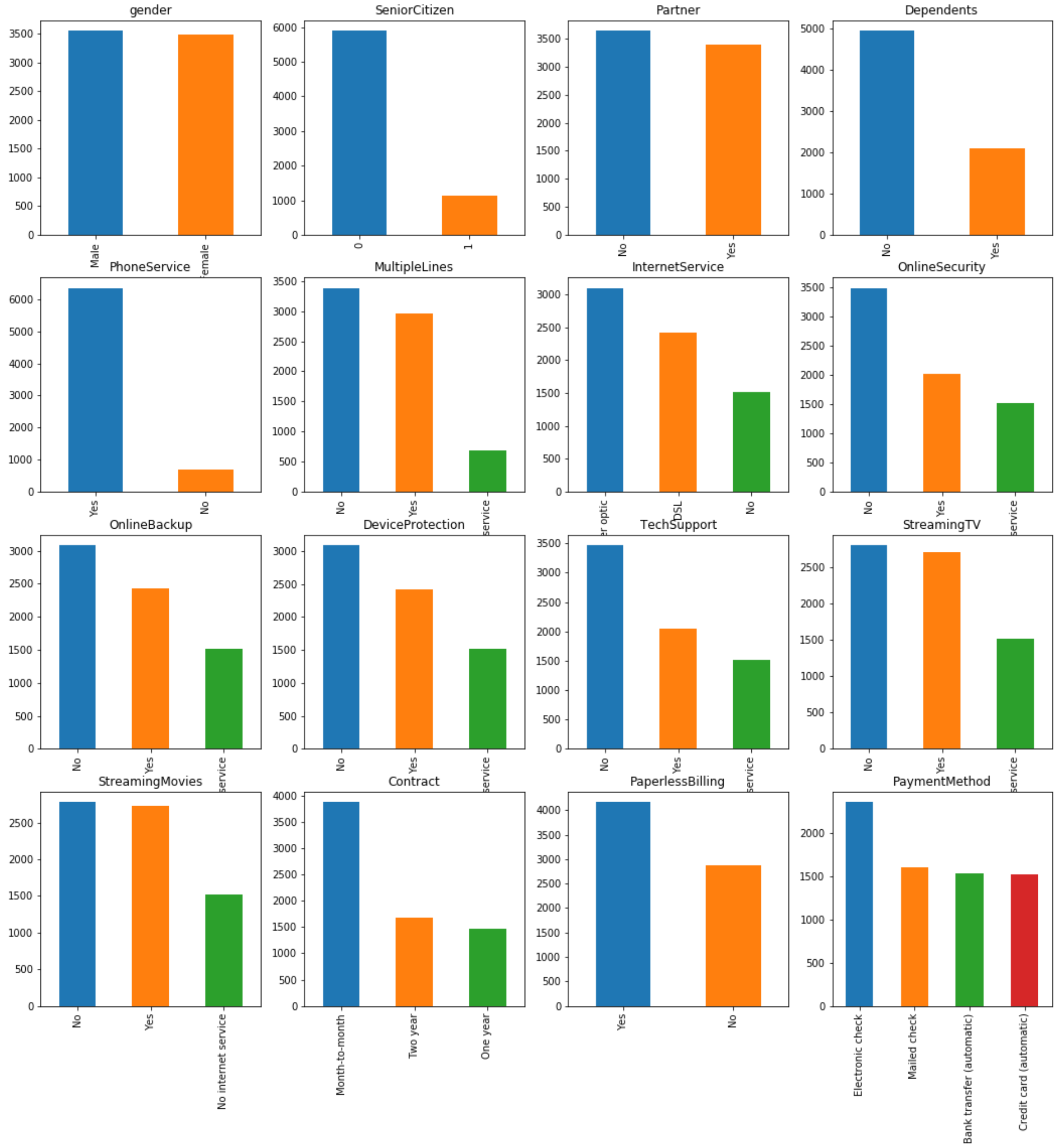
```
In [46]: fig, ax = plt.subplots(1, 3, figsize=(14, 4))
         df[df.Churn == "No"][numerical_features].hist(bins=30, color="blue", alpha=0.5, a
         x=ax)
         df[df.Churn == "Yes"][numerical_features].hist(bins=30, color="red", alpha=0.5, a
         x=ax)
```

```
Out[46]: array([<matplotlib.axes._subplots.AxesSubplot object at 0x00000259F2490978>,
                <matplotlib.axes._subplots.AxesSubplot object at 0x00000259F24C2710>,
                <matplotlib.axes._subplots.AxesSubplot object at 0x00000259F22CD710>],
               dtype=object)
```



In [47]:

```
ROWS, COLS = 4, 4
fig, ax = plt.subplots(ROWS, COLS, figsize=(18, 18))
row, col = 0, 0
for i, categorical_feature in enumerate(categorical_features):
    if col == COLS - 1:
        row += 1
    col = i % COLS
    df[categorical_feature].value_counts().plot('bar', ax=ax[row, col]).set_title
(categorical_feature)
```


gender
Male
emale
SeniorCitizen
Partner
No
Yes
Dependents
No
Yes
PhoneService
Yes
No
MultipleLines
No
Yes
service
InternetService
er optic
DSL
No
OnlineSecurity
No
Yes
service
OnlineBackup
No
Yes
service
DeviceProtection
No
Yes
service
TechSupport
No
Yes
service
StreamingTV
No
Yes
service
StreamingMovies
No
Yes
No internet service
Contract
Month-to-month
Two year
One year
PaperlessBilling
Yes
No
PaymentMethod
Electronic check
Mailed check
Bank transfer (automatic)
Credit card (automatic)