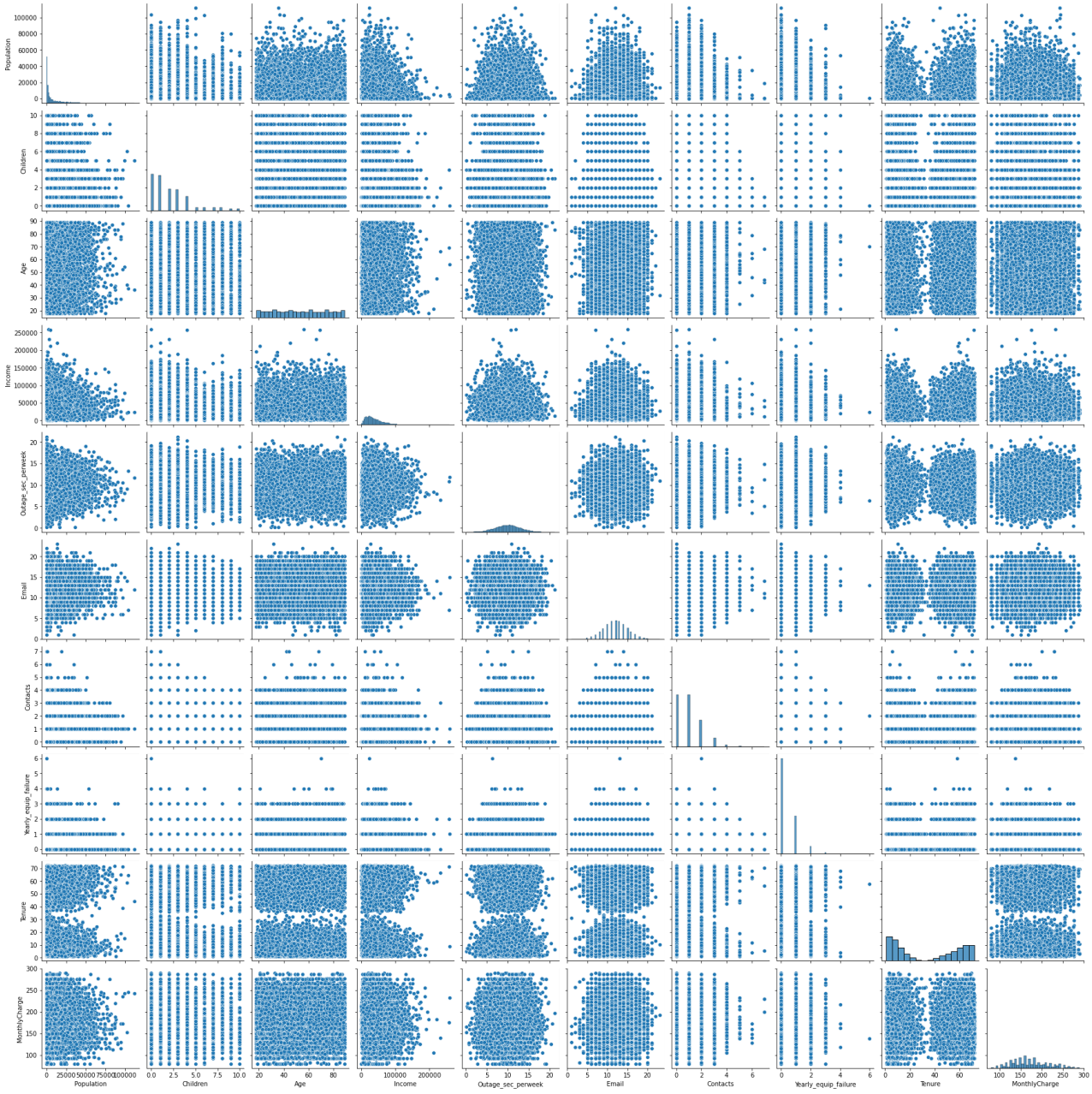
In [2]:
```
# Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Import dataset
churn_clean = pd.read_csv("C:/Users/hkeim/OneDrive/Documents/School/D212/churn_clean.csv")
```

In [3]:
```
# Select relevant continuous variables
variables = churn_clean.filter(items = ['Population', 'Children', 'Age', 'Income', 'Outage_se

# Check variables for variance
variables.describe()
```

Out[3]:

	Population	Children	Age	Income	Outage_sec_perweek	Email	Contacts
count	10000.000000	10000.0000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	9756.562400	2.0877	53.078400	39806.926771	10.001848	12.016000	0.994200
std	14432.698671	2.1472	20.698882	28199.916702	2.976019	3.025898	0.988466
min	0.000000	0.0000	18.000000	348.670000	0.099747	1.000000	0.000000
25%	738.000000	0.0000	35.000000	19224.717500	8.018214	10.000000	0.000000
50%	2910.500000	1.0000	53.000000	33170.605000	10.018560	12.000000	1.000000
75%	13168.000000	3.0000	71.000000	53246.170000	11.969485	14.000000	2.000000
max	111850.000000	10.0000	89.000000	258900.700000	21.207230	23.000000	7.000000

In [4]:
```
# Import seaborn
import seaborn as sns

# Visually check variables for variance
sns.pairplot(variables, diag_kind='hist')
plt.show()
```



In [5]:
```
# Drop redundant variables
reduced_variables = variables.drop(['Email', 'Contacts'], axis = 1)
```

In [6]:
```
# Normalize the data
normalized_variables = reduced_variables / reduced_variables.mean()

# Print the variances of the normalized data
print(normalized_variables.var())
```

```
Population             2.188273
Children               1.057813
Age                    0.152075
Income                 0.501855
Outage_sec_perweek     0.088534
Yearly_equip_failure   2.553196
Tenure                 0.586579
MonthlyCharge          0.061884
dtype: float64
```

In [7]:
```
# Drop variables with < 0.5 variance
reduced_variables = normalized_variables.drop(['Age', 'MonthlyCharge', 'Outage_sec_perweek'],
```

In [8]:
```
# Determine ratio of nulls in columns
percent_missing = reduced_variables.isnull().sum() * 100 / len(reduced_variables)
missing_value_df = pd.DataFrame({'percent_missing': percent_missing})
print(missing_value_df)
```

```
                     percent_missing
Population                       0.0
Children                         0.0
Income                           0.0
Yearly_equip_failure             0.0
Tenure                           0.0
```

In [9]:
```
# Get correlation matrix
reduced_variables.corr()
```

Out[9]:

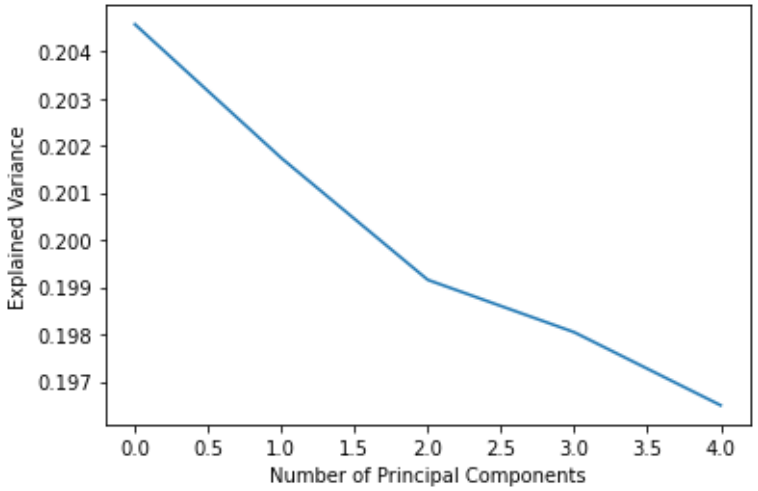
	Population	Children	Income	Yearly_equip_failure	Tenure
Population	1.000000	-0.005877	-0.008639	-0.004483	-0.003559
Children	-0.005877	1.000000	0.009942	0.007321	-0.005091
Income	-0.008639	0.009942	1.000000	0.005423	0.002114
Yearly_equip_failure	-0.004483	0.007321	0.005423	1.000000	0.012435
Tenure	-0.003559	-0.005091	0.002114	0.012435	1.000000

In [10]:
```
# Perform PCA

# Import libraries and packages
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.pipeline import Pipeline

# Get feature matrix
X = reduced_variables.values
print(X.shape)

# Scale data
scaler = StandardScaler()
scaler.fit(X)
X_scaled = scaler.transform(X)

# Run PCA
pca_5 = PCA()
pca_5.fit(X_scaled)
X_pca_5 = pca_5.transform(X_scaled)
print('Variance explained by all 5 principal components', sum(pca_5.explained_variance_ratio_
print(pca_5.explained_variance_ratio_ * 100)
print(np.cumsum(pca_5.explained_variance_ratio_ * 100))
```

```
(10000, 5)
Variance explained by all 5 principal components 100.00000000000001
[20.45689653 20.17357359 19.91545182 19.80448176 19.64959629]
[ 20.45689653  40.63047012  60.54592194  80.35040371 100.        ]
```

In [11]:
```
# Visualize components
```

```
plt.plot((pca_5.explained_variance_ratio_))
plt.xlabel('Number of Principal Components')
plt.ylabel('Explained Variance')
plt.savefig('elbowplot.png', dpi = 100)
```



In [15]:

```
loadings = pd.DataFrame(pca_5.components_.T,
columns=['PC1','PC2','PC3','PC4','PC5'],
index=reduced_variables.columns)
loadings
```

Out[15]:

	PC1	PC2	PC3	PC4	PC5
Population	-0.450657	0.118529	0.727113	0.498767	-0.073474
Children	0.434291	-0.497231	0.469973	-0.252458	-0.528718
Income	0.508293	-0.274749	-0.162289	0.784294	0.157150
Yearly_equip_failure	0.510735	0.379836	0.462487	-0.217950	0.577475
Tenure	0.298476	0.720382	-0.100998	0.157724	-0.597398

In [65]:

```
# Export reduced_variables for report purposes
reduced_variables.to_csv("C:/Users/hkeim/OneDrive/Documents/School/D212/Task Two/Keim D212 Ta
```

In [ ]: