In [13]:
```
import pandas as pd
from scipy.stats import chi2_contingency
from itertools import combinations
from scipy.stats import ttest_ind
import seaborn as sns
import matplotlib.pyplot as plt
```

In [28]:
```
df = pd.read_csv(r"WA_Fn-UseC_-Telco-Customer-Churn.csv")
```

In [29]:
```
df
```

Out[29]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	Mul
0	7590-VHVEG	Female	0	Yes	No	1	No	
1	5575-GNVDE	Male	0	No	No	34	Yes	
2	3668-QPYBK	Male	0	No	No	2	Yes	
3	7795-CFOCW	Male	0	No	No	45	No	
4	9237-HQITU	Female	0	No	No	2	Yes	
...	...	...	...	...	...	...	...	
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	
7040	4801-JZAZL	Female	0	Yes	Yes	11	No	
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	
7042	3186-AJIEK	Male	0	No	No	66	Yes	

7043 rows × 21 columns

Chi-Square Statistics.

A p-value lower than 0.05 indicates strong evidence against the null hypothesis. This means that if the p value is lower than 0.05, the variable being analyzed has a strong impact on the churn rate.

In [30]:

```
contingency_online_security = pd.crosstab(df['OnlineSecurity'], df['Churn'])
print(contingency_online_security)
```

```
Churn                  No   Yes
OnlineSecurity
No                   2037  1461
No internet service  1413   113
Yes                  1724   295
```

In [31]:

```
chi2, p, dof, ex = chi2_contingency(contingency_online_security)

print("Chi-Square Statistic:", chi2)
print("P-Value:", p)
print("Degrees of Freedom:", dof)
print("Expected Frequencies:")
print(ex)
```

```
Chi-Square Statistic: 849.9989679615965
P-Value: 2.66114963517655 17e-185
Degrees of Freedom: 2
Expected Frequencies:
[[2569.73619196  928.26380804]
 [1121.04557717  404.95442283]
 [1483.21823087  535.78176913]]
```

This p-value is 2.66*10^-185 < 0.05. There is a strong correlation between online security and churn.

In [32]:

```
contingency_gender = pd.crosstab(df['gender'], df['Churn'])
print(contingency_gender)

chi2, p, dof, ex = chi2_contingency(contingency_gender)

print("Chi-Square Statistic:", chi2)
print("P-Value:", p)
print("Degrees of Freedom:", dof)
print("Expected Frequencies:")
print(ex)
```

```
Churn     No  Yes
gender
Female  2549  939
Male    2625  930
Chi-Square Statistic: 0.4840828822091383
P-Value: 0.48657873605618596
Degrees of Freedom: 1
Expected Frequencies:
[[2562.38989067  925.61010933]
 [2611.61010933  943.38989067]]
```

This p-value is 0.49 > 0.05. There is not a correlation between gender and churn.

In [33]:

```
categorical_columns = df.select_dtypes(include=['object', 'category']).columns
categorical_columns = [col for col in df if col != 'Churn' and col != 'MonthlyCharg
```

In [34]:

```
categorical_columns
```

Out[34]:

```
['gender',
 'SeniorCitizen',
 'Partner',
 'Dependents',
 'PhoneService',
 'MultipleLines',
 'InternetService',
 'OnlineSecurity',
 'OnlineBackup',
 'DeviceProtection',
 'TechSupport',
 'StreamingTV',
 'StreamingMovies',
 'Contract',
 'PaperlessBilling',
 'PaymentMethod']
```

In [35]:

```
def chi_square_test_with_churn(df, categorical_columns):
    results = []
    for var in categorical_columns:
        # Create a contingency table
        contingency_table = pd.crosstab(df[var], df['Churn'])
        # Perform chi-square test
        chi2, p, dof, ex = chi2_contingency(contingency_table)
        results.append((var, chi2, p, dof, ex))
    # Sort results by p-value
    results.sort(key=lambda x: x[2])
    return results
results = chi_square_test_with_churn(df, categorical_columns)
```

In [36]:

```
for var, chi2, p, dof, ex in results:
    print(f"Chi-Square Test between {var} and Churn:")
    print(f"Chi-Square Statistic: {chi2}")
    print(f"P-Value: {p}")
    print(f"Degrees of Freedom: {dof}")
    print(f"Expected Frequencies: \n{ex}\n")
```

```
Chi-Square Test between Contract and Churn:
Chi-Square Statistic: 1184.5965720837926
P-Value: 5.863038300673391e-258
Degrees of Freedom: 2
Expected Frequencies:
[[2846.69175067 1028.30824933]
 [1082.11018032  390.88981968]
 [1245.198069    449.801931  ]]

Chi-Square Test between OnlineSecurity and Churn:
Chi-Square Statistic: 849.9989679615965
P-Value: 2.6611496351765517e-185
Degrees of Freedom: 2
Expected Frequencies:
[[2569.73619196  928.26380804]
 [1121.04557717  404.95442283]
 [1483.21823087  535.78176913]]

Chi-Square Test between TechSupport and Churn:
Chi-Square Statistic: 828.1970684587394
P-Value: 1.4430840279998987e-180
Degrees of Freedom: 2
Expected Frequencies:
[[2551.37043873  921.62956127]
 [1121.04557717  404.95442283]
 [1501.5839841   542.4160159 ]]

Chi-Square Test between InternetService and Churn:
Chi-Square Statistic: 732.309589667794
P-Value: 9.571788222840544e-160
Degrees of Freedom: 2
Expected Frequencies:
[[1778.53954281  642.46045719]
 [2274.41488002  821.58511998]
 [1121.04557717  404.95442283]]

Chi-Square Test between PaymentMethod and Churn:
Chi-Square Statistic: 648.1423274814
P-Value: 3.6823546520097993e-140
Degrees of Freedom: 3
Expected Frequencies:
[[1134.26891949  409.73108051]
 [1118.10705665  403.89294335]
 [1737.40025557  627.59974443]
 [1184.22376828  427.77623172]]

Chi-Square Test between OnlineBackup and Churn:
Chi-Square Statistic: 601.812790113409
P-Value: 2.0797592160864276e-131
Degrees of Freedom: 2
Expected Frequencies:
[[2268.53783899  819.46216101]
 [1121.04557717  404.95442283]
 [1784.41658384  644.58341616]]

Chi-Square Test between DeviceProtection and Churn:
```

```
Chi-Square Statistic: 558.419369407389
P-Value: 5.505219496457244e-122
Degrees of Freedom: 2
Expected Frequencies:
[[2273.68024989  821.31975011]
 [1121.04557717  404.95442283]
 [1779.27417294  642.72582706]]

Chi-Square Test between StreamingMovies and Churn:
Chi-Square Statistic: 375.6614793452656
P-Value: 2.667756755723681e-82
Degrees of Freedom: 2
Expected Frequencies:
[[2045.94490984  739.05509016]
 [1121.04557717  404.95442283]
 [2007.00951299  724.99048701]]

Chi-Square Test between StreamingTV and Churn:
Chi-Square Statistic: 374.2039433109813
P-Value: 5.528994485739183e-82
Degrees of Freedom: 2
Expected Frequencies:
[[2064.31066307  745.68933693]
 [1121.04557717  404.95442283]
 [1988.64375976  718.35624024]]

Chi-Square Test between PaperlessBilling and Churn:
Chi-Square Statistic: 258.27764906707307
P-Value: 4.073354668665985e-58
Degrees of Freedom: 1
Expected Frequencies:
[[2109.85773108  762.14226892]
 [3064.14226892 1106.85773108]]

Chi-Square Test between Dependents and Churn:
Chi-Square Statistic: 189.12924940423474
P-Value: 4.9249216612154196e-43
Degrees of Freedom: 1
Expected Frequencies:
[[3623.93042737 1309.06957263]
 [1550.06957263  559.93042737]]

Chi-Square Test between SeniorCitizen and Churn:
Chi-Square Statistic: 159.42630036838742
P-Value: 1.510066805092378e-36
Degrees of Freedom: 1
Expected Frequencies:
[[4335.05239245 1565.94760755]
 [ 838.94760755  303.05239245]]

Chi-Square Test between Partner and Churn:
Chi-Square Statistic: 158.7333820309922
P-Value: 2.1399113440759935e-36
Degrees of Freedom: 1
Expected Frequencies:
[[2674.78830044  966.21169956]
```

```
 [2499.21169956  902.78830044]]

Chi-Square Test between MultipleLines and Churn:
Chi-Square Statistic: 11.33044148319756
P-Value: 0.0034643829548773
Degrees of Freedom: 2
Expected Frequencies:
[[2490.39613801  899.60386199]
 [ 501.01774812  180.98225188]
 [2182.58611387  788.41388613]]

Chi-Square Test between PhoneService and Churn:
Chi-Square Statistic: 0.9150329892546948
P-Value: 0.3387825358066928
Degrees of Freedom: 1
Expected Frequencies:
[[ 501.01774812  180.98225188]
 [4672.98225188 1688.01774812]]

Chi-Square Test between gender and Churn:
Chi-Square Statistic: 0.4840828822091383
P-Value: 0.48657873605618596
Degrees of Freedom: 1
Expected Frequencies:
[[2562.38989067  925.61010933]
 [2611.61010933  943.38989067]]
```

In [37]:

```
numerical_columns = [col for col in df if col == 'tenure' or col == 'MonthlyCharges
```

In [38]:

```
numerical_columns
```

Out[38]:

```
['tenure', 'MonthlyCharges']
```

In [39]:

```
def t_test_with_churn(df, numerical_columns):
    results = []
    for var in numerical_columns:
        # Separate the data into two groups based on the 'Churn' column
        group1 = df[df['Churn'] == 'Yes'][var]
        group2 = df[df['Churn'] == 'No'][var]
        # Perform t-test
        t_stat, p_val = ttest_ind(group1, group2, nan_policy='omit')
        results.append((var, t_stat, p_val))
    # Sort results by p-value
    results.sort(key=lambda x: x[2])
    return results
results = t_test_with_churn(df, numerical_columns)
```

In [40]:

```
df.dtypes
```

Out[40]:

```
customerID           object
gender               object
SeniorCitizen         int64
Partner              object
Dependents           object
tenure                int64
PhoneService         object
MultipleLines        object
InternetService      object
OnlineSecurity       object
OnlineBackup         object
DeviceProtection     object
TechSupport          object
StreamingTV          object
StreamingMovies      object
Contract             object
PaperlessBilling     object
PaymentMethod        object
MonthlyCharges      float64
TotalCharges         object
Churn                object
dtype: object
```

In [41]:

```
for var, t_stat, p_val in results:
    print(f"T-Test between {var} and Churn:")
    print(f"T-Statistic: {t_stat}")
    print(f"P-Value: {p_val}\n")
```

```
T-Test between tenure and Churn:
T-Statistic: -31.57955051135377
P-Value: 7.99905796059022e-205

T-Test between MonthlyCharges and Churn:
T-Statistic: 16.536738015936308
P-Value: 2.7066456068884154e-60
```

In [42]:

```
non_numeric_total_charges = df[pd.to_numeric(df['TotalCharges'], errors='coerce').i
print("Non-numeric TotalCharges values:")
print(non_numeric_total_charges)
```

```
Non-numeric TotalCharges values:
      customerID  gender  SeniorCitizen Partner Dependents  tenure
488   4472-LVYGI  Female              0     Yes        Yes       0  \
753   3115-CZMZD    Male              0      No        Yes       0
936   5709-LVOEQ  Female              0     Yes        Yes       0
1082  4367-NUYAO    Male              0     Yes        Yes       0
1340  1371-DWPAZ  Female              0     Yes        Yes       0
3331  7644-OMVMY    Male              0     Yes        Yes       0
3826  3213-VVOLG    Male              0     Yes        Yes       0
4380  2520-SGTTA  Female              0     Yes        Yes       0
5218  2923-ARZLG    Male              0     Yes        Yes       0
6670  4075-WKNIU  Female              0     Yes        Yes       0
6754  2775-SEFEE    Male              0      No        Yes       0

     PhoneService     MultipleLines InternetService       OnlineSecurity  ...
488            No  No phone service             DSL                  Yes  ...  \
753           Yes                No              No  No internet service  ...
936           Yes                No             DSL                  Yes  ...
1082          Yes               Yes              No  No internet service  ...
1340           No  No phone service             DSL                  Yes  ...
3331          Yes                No              No  No internet service  ...
3826          Yes               Yes              No  No internet service  ...
4380          Yes                No              No  No internet service  ...
5218          Yes                No              No  No internet service  ...
6670          Yes               Yes             DSL                   No  ...
6754          Yes               Yes             DSL                  Yes  ...

         DeviceProtection          TechSupport          StreamingTV
488                   Yes                  Yes                  Yes  \
753   No internet service  No internet service  No internet service
936                   Yes                   No                  Yes
1082  No internet service  No internet service  No internet service
1340                  Yes                  Yes                  Yes
3331  No internet service  No internet service  No internet service
3826  No internet service  No internet service  No internet service
4380  No internet service  No internet service  No internet service
5218  No internet service  No internet service  No internet service
6670                  Yes                  Yes                  Yes
6754                   No                  Yes                   No

          StreamingMovies  Contract PaperlessBilling
488                    No  Two year              Yes  \
753   No internet service  Two year               No
936                   Yes  Two year               No
1082  No internet service  Two year               No
1340                   No  Two year               No
3331  No internet service  Two year               No
3826  No internet service  Two year               No
4380  No internet service  Two year               No
5218  No internet service  One year              Yes
6670                   No  Two year               No
6754                   No  Two year              Yes

                 PaymentMethod MonthlyCharges  TotalCharges Churn
488   Bank transfer (automatic)          52.55                  No
753                Mailed check          20.25                  No
```

```
936                 Mailed check          80.85                   No
1082                Mailed check          25.75                   No
1340     Credit card (automatic)          56.05                   No
3331                Mailed check          19.85                   No
3826                Mailed check          25.35                   No
4380                Mailed check          20.00                   No
5218                Mailed check          19.70                   No
6670                Mailed check          73.35                   No
6754   Bank transfer (automatic)          61.90                   No

[11 rows x 21 columns]
```

In [43]:

```python
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')

# Step 5: Handle missing values if necessary (e.g., fill with 0 or drop)
# Here, we fill NaN values with 0
df['TotalCharges'].fillna(0, inplace=True)

# Step 6: Verify the conversion
print(df['TotalCharges'].dtype)
```

```
float64
```

In [44]:

```python
churn_yes = df[df['Churn'] == 'Yes']['TotalCharges']
churn_no = df[df['Churn'] == 'No']['TotalCharges']
t_stat, p_val = ttest_ind(churn_yes, churn_no, nan_policy='omit')

# Step 6: Display the results
print(f"T-Test between TotalCharges and Churn:")
print(f"T-Statistic: {t_stat}")
print(f"P-Value: {p_val}")
```

```
T-Test between TotalCharges and Churn:
T-Statistic: -16.978779727124437
P-Value: 2.127211613240394e-63
```

In [45]:

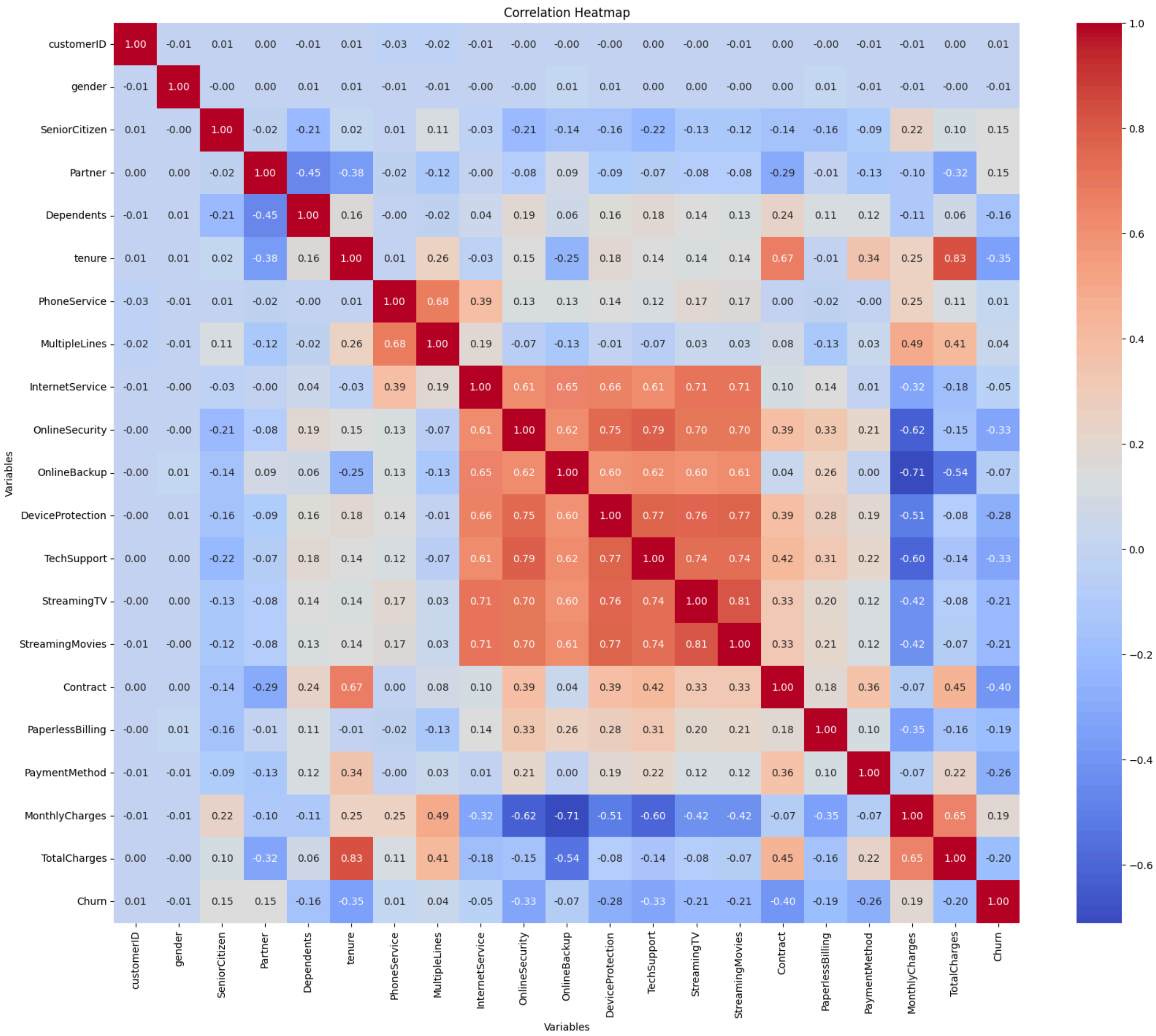
```python
categorical_columns = df.select_dtypes(include=['object', 'category']).columns

for col in categorical_columns:
    df[col] = pd.factorize(df[col])[0]

corr_matrix = df.corr()

plt.figure(figsize=(20, 16))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap')
plt.xlabel('Variables')
plt.ylabel('Variables')

plt.savefig('heatmap.png')
```



	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
customerID	1.00	-0.01	0.01	0.00	-0.01	0.01	-0.03	-0.02	-0.01	-0.00	-0.00	-0.00	0.00	-0.00	-0.01	0.00	-0.00	-0.01	-0.01	0.00	0.01
gender	-0.01	1.00	-0.00	0.00	0.01	0.01	-0.01	-0.01	-0.00	-0.00	0.01	0.01	0.00	0.00	-0.00	0.00	0.01	-0.01	-0.01	-0.00	-0.01
SeniorCitizen	0.01	-0.00	1.00	-0.02	-0.21	0.02	0.01	0.11	-0.03	-0.21	-0.14	-0.16	-0.22	-0.13	-0.12	-0.14	-0.16	-0.09	0.22	0.10	0.15
Partner	0.00	0.00	-0.02	1.00	-0.45	-0.38	-0.02	-0.12	-0.00	-0.08	0.09	-0.09	-0.07	-0.08	-0.08	-0.29	-0.01	-0.13	-0.10	-0.32	0.15
Dependents	-0.01	0.01	-0.21	-0.45	1.00	0.16	-0.00	-0.02	0.04	0.19	0.06	0.16	0.18	0.14	0.13	0.24	0.11	0.12	-0.11	0.06	-0.16
tenure	0.01	0.01	0.02	-0.38	0.16	1.00	0.01	0.26	-0.03	0.15	-0.25	0.18	0.14	0.14	0.14	0.67	-0.01	0.34	0.25	0.83	-0.35
PhoneService	-0.03	-0.01	0.01	-0.02	-0.00	0.01	1.00	0.68	0.39	0.13	0.13	0.14	0.12	0.17	0.17	0.00	-0.02	-0.00	0.25	0.11	0.01
MultipleLines	-0.02	-0.01	0.11	-0.12	-0.02	0.26	0.68	1.00	0.19	-0.07	-0.13	-0.01	-0.07	0.03	0.03	0.08	-0.13	0.03	0.49	0.41	0.04
InternetService	-0.01	-0.00	-0.03	-0.00	0.04	-0.03	0.39	0.19	1.00	0.61	0.65	0.66	0.61	0.71	0.71	0.10	0.14	0.01	-0.32	-0.18	-0.05
OnlineSecurity	-0.00	-0.00	-0.21	-0.08	0.19	0.15	0.13	-0.07	0.61	1.00	0.62	0.75	0.79	0.70	0.70	0.39	0.33	0.21	-0.62	-0.15	-0.33
OnlineBackup	-0.00	0.01	-0.14	0.09	0.06	-0.25	0.13	-0.13	0.65	0.62	1.00	0.60	0.62	0.60	0.61	0.04	0.26	0.00	-0.71	-0.54	-0.07
DeviceProtection	-0.00	0.01	-0.16	-0.09	0.16	0.18	0.14	-0.01	0.66	0.75	0.60	1.00	0.77	0.76	0.77	0.39	0.28	0.19	-0.51	-0.08	-0.28
TechSupport	0.00	0.00	-0.22	-0.07	0.18	0.14	0.12	-0.07	0.61	0.79	0.62	0.77	1.00	0.74	0.74	0.42	0.31	0.22	-0.60	-0.14	-0.33
StreamingTV	-0.00	0.00	-0.13	-0.08	0.14	0.14	0.17	0.03	0.71	0.70	0.60	0.76	0.74	1.00	0.81	0.33	0.20	0.12	-0.42	-0.08	-0.21
StreamingMovies	-0.01	-0.00	-0.12	-0.08	0.13	0.14	0.17	0.03	0.71	0.70	0.61	0.77	0.74	0.81	1.00	0.33	0.21	0.12	-0.42	-0.07	-0.21
Contract	0.00	0.00	-0.14	-0.29	0.24	0.67	0.00	0.08	0.10	0.39	0.04	0.39	0.42	0.33	0.33	1.00	0.18	0.36	-0.07	0.45	-0.40
PaperlessBilling	-0.00	0.01	-0.16	-0.01	0.11	-0.01	-0.02	-0.13	0.14	0.33	0.26	0.28	0.31	0.20	0.21	0.18	1.00	0.10	-0.35	-0.16	-0.19
PaymentMethod	-0.01	-0.01	-0.09	-0.13	0.12	0.34	-0.00	0.03	0.01	0.21	0.00	0.19	0.22	0.12	0.12	0.36	0.10	1.00	-0.07	0.22	-0.26
MonthlyCharges	-0.01	-0.01	0.22	-0.10	-0.11	0.25	0.25	0.49	-0.32	-0.62	-0.71	-0.51	-0.60	-0.42	-0.42	-0.07	-0.35	-0.07	1.00	0.65	0.19
TotalCharges	0.00	-0.00	0.10	-0.32	0.06	0.83	0.11	0.41	-0.18	-0.15	-0.54	-0.08	-0.14	-0.08	-0.07	0.45	-0.16	0.22	0.65	1.00	-0.20
Churn	0.01	-0.01	0.15	0.15	-0.16	-0.35	0.01	0.04	-0.05	-0.33	-0.07	-0.28	-0.33	-0.21	-0.21	-0.40	-0.19	-0.26	0.19	-0.20	1.00