

데이터 분석 실습(1차)


2018.11.

DT 추진 진행사항

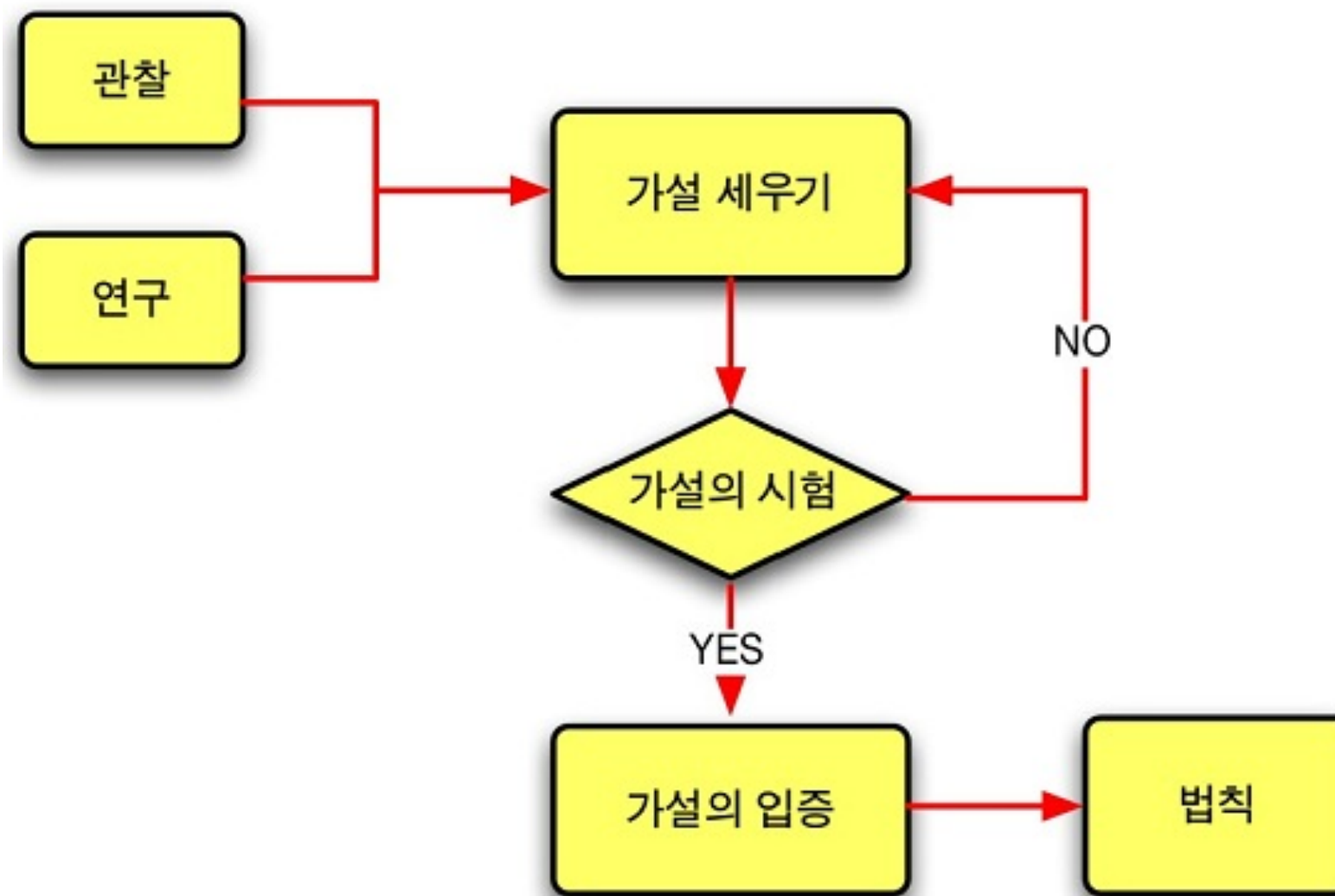
- 게시판: face.sktelink.com
- Quick-win과제
- 데이터 분석 실습(11/9(금), 12(월), 14(수), 16(금))

	11/9(금)	11/12(월)	11/14(수)	11/16(금)
오전 (9시~13시)	Infra(보라매)	기사본(남산)	경영지원(남산)	M본부(남산)
오후 (14시~18시)	Infra(보라매)	기사본(남산)	경영지원(남산)	M본부(남산)

분석 Tool 설치

- 아나콘다 설치 : www.google.com > anaconda 
 1. usb 파일 전체를 “ C:\사용자\skt\사번 “ 에 복사
 2. 설치 파일 실행, 모두 next (기본설정)으로 진행
- Jupyter notebook
 1. Python3 실행 → 파일이름변경 → Cell type(code/markdown)
 2. 자동완성 : tab
 3. 객체/함수 정보 조회 : shift + tab

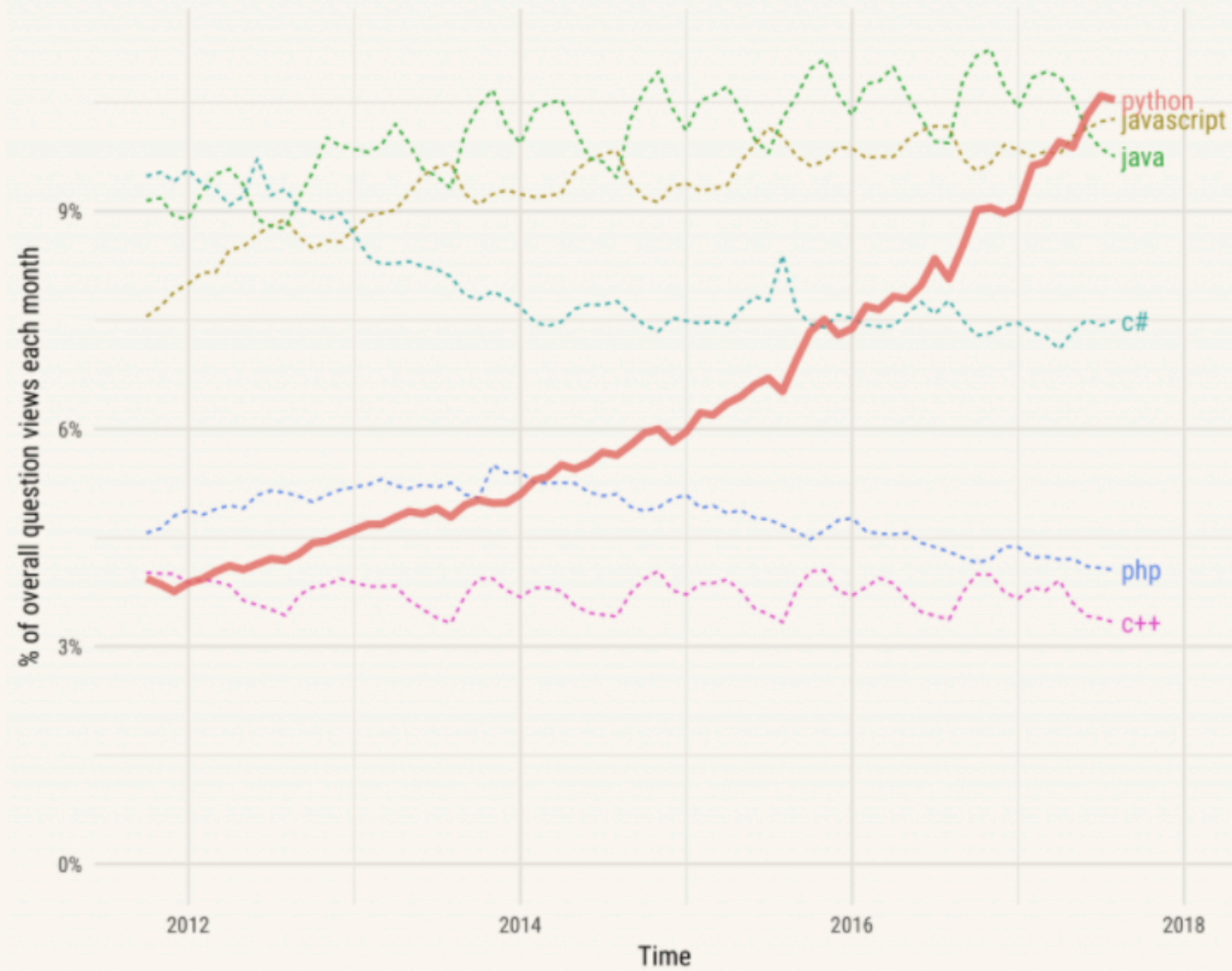
“ 가설 연역 방법은 오랜 경험주의적 사고를 바탕으로 등장한 과학적 연구의 방법이다. “
- Brody, Thomas A



[가설 연역 방법]
< 출처 : 위키피디아 >

Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries



파이썬 기본 문법

1. 문자열, 사칙연산(+ - * / % **), 논리연산(and or == != not), Comment¶

```
In [35]: print('Hello world') # print "hello world"

In [55]: 6+1 * 3, (6+1) * 3, 5/2, 5/2.0 # 사칙연산

In [57]: print('Hello'+ ' world') # print('Hello'+ ' world' *3)

In [74]: print("{} , {} , {} , {}".format(1==1, 1!=1, (1==1) and (1!=1), (1==1) or (1!=1))) # 논리연산
```

2 변수와 타입¶

```
In [1]: x = 5
In [2]: print(x,type(x))

In [3]: x = "Python"
In [4]: print(x,type(x))

In [5]: print(1==1 and 2!=1)
```

3 Python 데이터 타입¶

- 기본타입 : 숫자, 문자형, Bool, List, Tuple, Dictionary, File, Dates, Times, 함수, Class 등

4 조건문(if elif else), 반복문(for while)

```
In [82]: i = 3
         if (i%2 == 0):
             print('{} is even'.format(i))
         else:
             print('{} is odd'.format(i))
```

```
In [79]: for i in range(10):
         print(i)
```

```
In [83]: for i in range(10,15):
         if (i%2 == 0):
             print('{} is even'.format(i))
```

[연습]

- 0 ~ 100 까지 짝수를 출력하세요
☞ 0 2 4 6 ... 98 100
- 100 ~ 1 까지 출력하세요
☞ 100 99 98 ... 2 1

5 함수, Class

```
In [84]: def even_print(a,b):
         for i in range(a,b):
             if (i%2 == 0):
                 print('{} is even'.format(i))
```

```
In [85]: even_print(11,19)
```

6 모듈(라이브러리) 사용 : import, from

```
In [1]: import numpy as np
         import pandas as pd
```

라이브러리

Matplotlib, Seaborn, Numpy, Pandas

Matplotlib, Seaborn : 시각화(그래프)

```
In [90]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
In [91]: x = np.linspace(0, 2*np.pi, 500)
plt.plot(x, np.sin(x**2))
plt.title('test')
```

```
In [92]: x = np.arange(50)*2*np.pi/50
y = np.sin(x)
plt.plot(x,y)
plt.xlabel('index')
```

```
In [11]: plt.plot(x,y,x*2,y)
```

```
In [94]: plt.scatter(x,y)
```

```
In [12]: plt.bar(x*50,abs(y),2)
```

```
In [10]: plt.hist(np.random.randn(100))
```


Numpy : 고성능 과학 계산¶

```
import numpy as np
```

- *array zeros ones full , arange linspace , random*¶

```
In [123]: import numpy as np
In [125]: a1 = np.array([1,2,3])
In [126]: a2 = np.array([4,5,6])
In [132]: a1+a2                                # 사칙연산
In [133]: a3 = np.zeros(10)
In [139]: a4 = np.ones((3,3))
In [141]: a5 = np.full((3,4), 3.14)
In [144]: a1.shape
In [145]: a5.shape
In [148]: a6 = np.arange(0, 20, 2)
In [156]: a7 = np.linspace(0, 1, 5)
In [161]: a8 = np.random.random((3, 3))        # normal , randint
```

- *1,2,3 .. n차원 배열*¶

```
In [165]: x1 = np.random.randint(10, size = 6)
In [167]: x1.shape
In [192]: x2 = np.random.randint(10, size = (3, 4))
In [194]: x2.shape
In [195]: x3 = np.random.randint(10, size = (3, 4, 5))
In [197]: x3.shape
In [198]: x3[0][2][2]
In [199]: x3[1:]
In [200]: x3[1,1,3:]
In [209]: x2[1,:]
In [205]: x2[:-1,:-1]
In [211]: np.reshape(x2,(4,3))
In [213]: np.reshape(x2,(2,6))
In [214]: # 기타 : newaxis, concatenate, vstack / hstack, split, vsplit / hsplit 등등
```

Pandas : 데이터프레임¶

```
import pandas as pd
```

```
In [2]: path = './score.csv'  
        score = pd.read_csv(path)
```

```
In [3]: score
```

```
In [4]: path = './excel_data.xlsx'  
        data = pd.read_excel(path, skiprows=1, usecols=4)
```

```
In [5]: data
```

```
In [6]: data.head(2)
```

```
In [7]: data.shape
```

```
In [8]: data.describe()
```

```
In [9]: data.describe().T
```

```
In [10]: data.columns
```

```
In [11]: data.index
```

```
In [12]: data[['B', 'E', 'D']].head(2)
```

```
In [13]: data.loc[2:4]
```

```
In [14]: data.loc[2:4, ['B', 'D']]
```

샘플:통신회사 고객이탈

- **customerID** - 고객 ID : 고객 ID
- **gender** - 성별 : 고객 성별 (여성, 남성)
- **SeniorCitizen** - 고령자 : 고객이 노약자인지 여부 (1, 0)
- **Partner** - 파트너 : 고객에게 파트너가 있는지 여부 (예, 아니오)
- **Dependents** - 부양 가족 : 고객이 부양 가족 여부 (예, 아니오)
- **tenure** - 보유 : 고객이 회사에 머물렀던 개월 수
- **PhoneService** - PhoneService : 고객에게 전화 서비스가 있는지 여부 (예, 아니오)
- **MultipleLines** - 다중 회선 : 고객이 여러 회선을 사용하는지 여부 (예, 아니오, 전화 서비스 없음)
- **InternetService** - 인터넷 서비스 : 고객의 인터넷 서비스 제공 업체 (DSL, 광섬유, 아니오)
- **OnlineSecurity** - 온라인 보안 : 고객의 온라인 보안 여부 (예, 아니오, 인터넷 서비스 없음)
- **OnlineBackup** - 온라인 백업 : 고객이 온라인 백업을했는지 여부 (예, 아니오, 인터넷 서비스 없음)
- **DeviceProtection** - DeviceProtection : 고객에게 기기 보호 기능이 있는지 여부 (예, 아니오, 인터넷 서비스 없음)
- **TechSupport** - 기술 지원 : 고객이 기술 지원을 받았는지 여부 (예, 아니오, 인터넷 서비스 없음)
- **StreamingTV** - 스트리밍 TV : 고객이 스트리밍 TV를 가지고 있는지 여부 (예, 아니오, 인터넷 서비스 없음)
- **StreamingMovies** - 스트리밍 영화 : 고객이 영화를 스트리밍하는지 여부 (예, 아니오, 인터넷 서비스 없음)
- **Contract** - 계약 : 고객의 계약 기간 (월간, 1 년, 2 년)
- **PaperlessBilling** - 페이퍼리스 결제 : 고객이 종이없는 청구서 수신 여부 (예, 아니오)
- **PaymentMethod** - PaymentMethod : 고객의 결제 수단 (전자 수표, 우편 수표, 은행 송금 (자동), 신용 카드 (자동))
- **MonthlyCharges** - 월별 요금 : 매월 고객에게 청구되는 금액
- **TotalCharges** - 총 요금 : 고객에게 청구 된 총 금액
- **Churn** - 이탈 : 고객의 이탈 여부 (예 또는 아니오)

```
In [6]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
In [7]: telco = pd.read_csv('./telco.csv')
telch = pd.read_csv('./telch.csv')
```

► In [8]: telch.head(5)

Out[8]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity
0	7590-VHVEG	Female	0	1	0	1	0	0 phone service	DSL	0
1	5575-GNVDE	Male	0	0	0	34	1	0	DSL	1
2	3668-QPYBK	Male	0	0	0	2	1	0	DSL	1
3	7795-CFOCW	Male	0	0	0	45	0	0 phone service	DSL	1
4	9237-HQITU	Female	0	0	0	2	1	0	Fiber optic	0

5 rows × 21 columns

```
In [9]: telch[['tenure', 'MonthlyCharges', 'TotalCharges', 'Churn', 'Contract', 'SeniorCitizen']].tail(5)
```

Out[9]:

	tenure	MonthlyCharges	TotalCharges	Churn	Contract	SeniorCitizen
7038	24	84.80	1990.5	0	One year	0
7039	72	103.20	7362.9	0	One year	0
7040	11	29.60	346.45	0	Month-to-month	0
7041	4	74.40	306.6	1	Month-to-month	1
7042	66	105.65	6844.5	0	Two year	0

```
In [10]: telch.shape
```

Out[10]: (7043, 21)



```
In [11]: for item in telch.columns:  
         print(item)  
         print(telch[item].unique())
```

```
customerID  
['7590-VHVEG' '5575-GNVDE' '3668-QPYBK' ... '4801-JZAZL' '8361-LTMKD'  
 '3186-AJIEK']  
gender  
['Female' 'Male']  
SeniorCitizen  
[0 1]  
Partner  
[1 0]  
Dependents  
[0 1]  
tenure  
[ 1 34  2 45  8 22 10 28 62 13 16 58 49 25 69 52 71 21 12 30 47 72 17 27  
  5 46 11 70 63 43 15 60 18 66  9  3 31 50 64 56  7 42 35 48 29 65 38 68  
 32 55 37 36 41  6  4 33 67 23 57 61 14 20 53 40 59 24 44 19 54 51 26  0  
 39]  
PhoneService  
[0 1]  
MultipleLines  
['0 phone service' '0' '1']  
InternetService  
['DSL' 'Fiber optic' '0']  
OnlineSecurity  
['0' '1' '0 internet service']  
OnlineBackup  
['1' '0' '0 internet service']  
DeviceProtection  
['0' '1' '0 internet service']  
TechSupport  
['0' '1' '0 internet service']  
StreamingTV  
['0' '1' '0 internet service']  
StreamingMovies  
['0' '1' '0 internet service']  
Contract  
['Month-to-month' 'One year' 'Two year']  
PaperlessBilling  
[1 0]  
PaymentMethod  
['Electronic check' 'Mailed check' 'Bank transfer (automatic)'  
 'Credit card (automatic)']  
MonthlyCharges  
[29.85 56.95 53.85 ... 63.1 44.2 78.7 ]  
TotalCharges  
['29.85' '1889.5' '108.15' ... '346.45' '306.6' '6844.5']  
Columns
```

...



In [12]: telch.info()

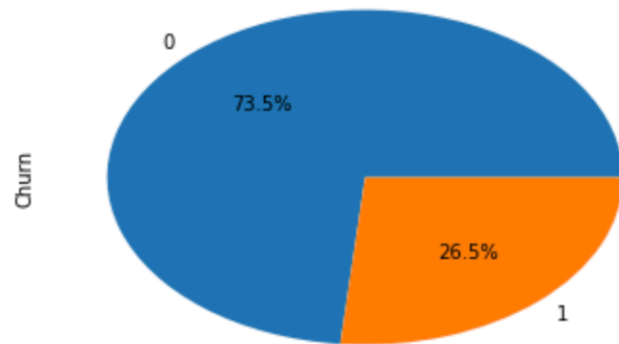
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
customerID    7043 non-null object
gender        7043 non-null object
SeniorCitizen 7043 non-null int64
Partner       7043 non-null int64
Dependents    7043 non-null int64
tenure        7043 non-null int64
PhoneService  7043 non-null int64
MultipleLines 7043 non-null object
InternetService 7043 non-null object
OnlineSecurity 7043 non-null object
OnlineBackup  7043 non-null object
DeviceProtection 7043 non-null object
TechSupport   7043 non-null object
StreamingTV   7043 non-null object
StreamingMovies 7043 non-null object
Contract      7043 non-null object
PaperlessBilling 7043 non-null int64
PaymentMethod 7043 non-null object
MonthlyCharges 7043 non-null float64
TotalCharges  7043 non-null object
Churn         7043 non-null int64
dtypes: float64(1), int64(7), object(13)
memory usage: 1.1+ MB
```

In [13]: `telch['TotalCharges'] = telch['TotalCharges'].replace(r'\s+', np.nan, regex = True)`
`telch['TotalCharges'] = pd.to_numeric(telch['TotalCharges'])`

In [14]: telch.info()

```
In [306]: telch['Churn'].value_counts().plot.pie(autopct = '%1.1f%%')
```

```
Out[306]: <matplotlib.axes._subplots.AxesSubplot at 0x1a269736d8>
```



```
In [307]: sns.factorplot(x = 'Contract', y = 'MonthlyCharges', hue = 'PaymentMethod', kind = 'point', data = telch)
```

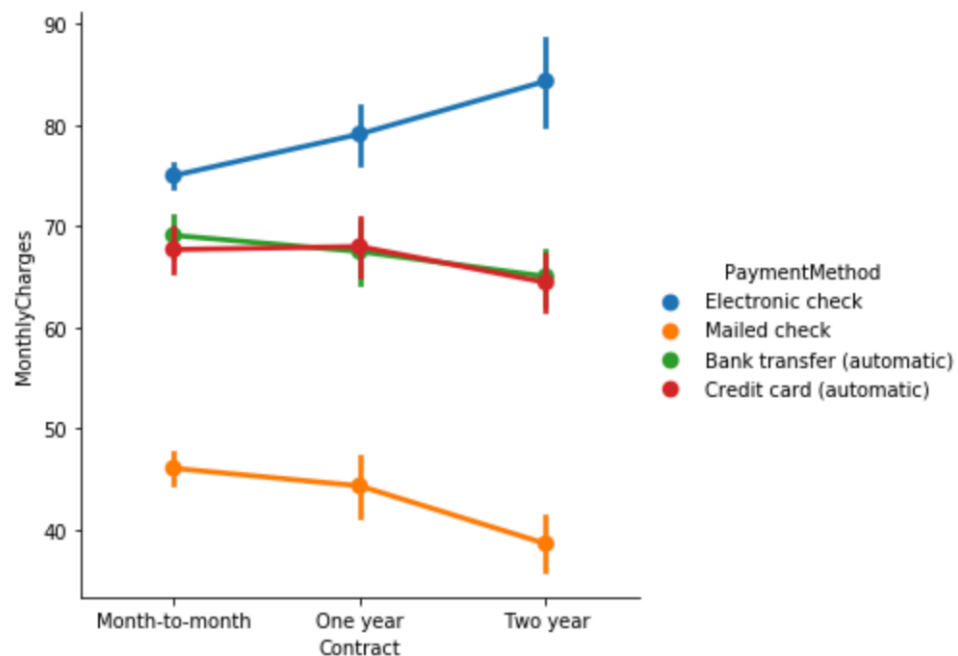
/anaconda3/lib/python3.7/site-packages/seaborn/categorical.py:3666: UserWarning: The `factorplot` function has been renamed to `catplot`. The original name will be removed in a future release. Please update your code. Note that the default `kind` in `factorplot` (`'point'`) has changed to `strip` in `catplot`.

warnings.warn(msg)

/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a different result.

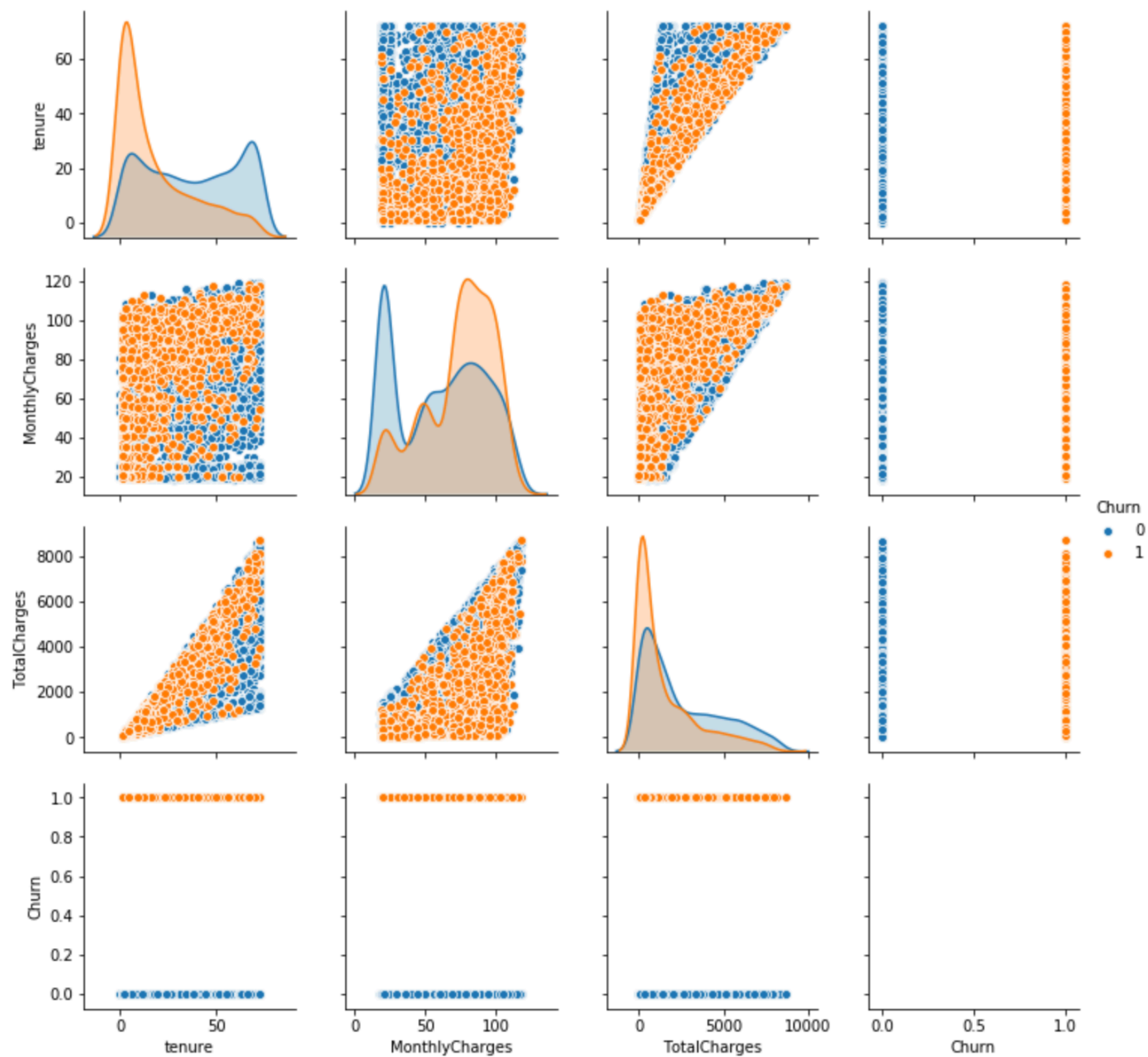
return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

```
Out[307]: <seaborn.axisgrid.FacetGrid at 0x1a270bd978>
```



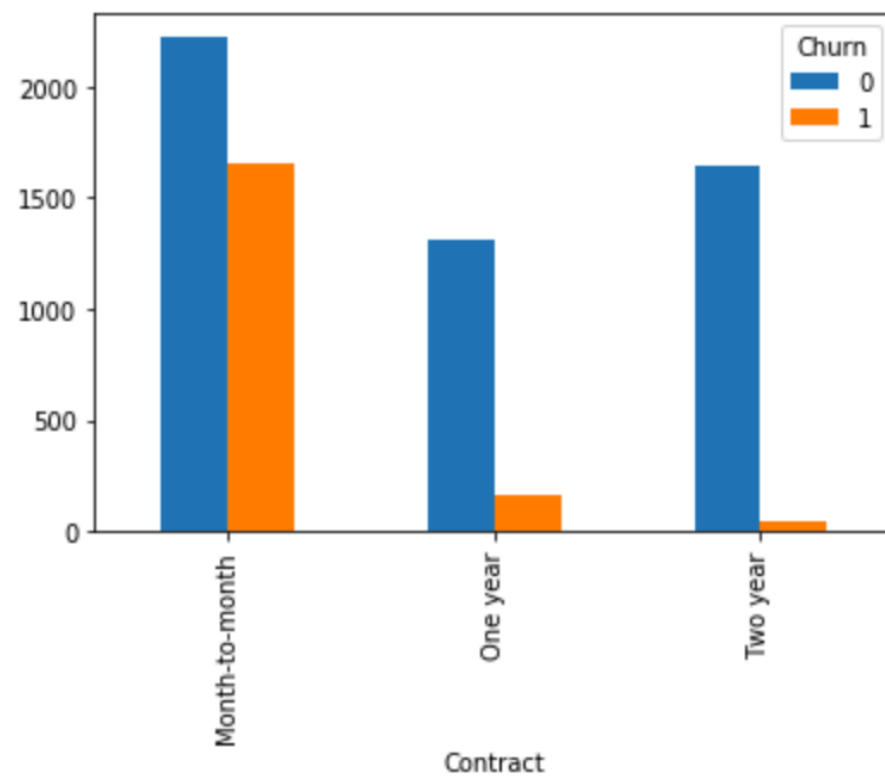
```
In [308]: sns.pairplot(telch[['tenure', 'MonthlyCharges', 'TotalCharges', 'Churn']], hue = 'Churn')
```

```
Out[308]: <seaborn.axisgrid.PairGrid at 0x1a27ce0390>
```




```
In [309]: pd.crosstab(telch.Contract, telch.Churn).plot(kind = 'bar')
```

```
Out[309]: <matplotlib.axes._subplots.AxesSubplot at 0x1a28436320>
```



```
In [310]: telch.describe()
```

Out[310]:

	SeniorCitizen	Partner	Dependents	tenure	PhoneService	PaperlessBilling	MonthlyCharges	TotalCharges	Churn
count	7043.000000	7043.000000	7043.000000	7043.000000	7043.000000	7043.000000	7043.000000	7032.000000	7043.000000
mean	0.162147	0.483033	0.299588	32.371149	0.903166	0.592219	64.761692	2283.300441	0.162147
std	0.368612	0.499748	0.458110	24.559481	0.295752	0.491457	30.090047	2266.771362	0.368612
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	18.250000	18.800000	0.000000
25%	0.000000	0.000000	0.000000	9.000000	1.000000	0.000000	35.500000	401.450000	0.000000
50%	0.000000	0.000000	0.000000	29.000000	1.000000	1.000000	70.350000	1397.475000	0.000000
75%	0.000000	1.000000	1.000000	55.000000	1.000000	1.000000	89.850000	3794.737500	0.000000
max	1.000000	1.000000	1.000000	72.000000	1.000000	1.000000	118.750000	8684.800000	1.000000

```
In [315]: sns.heatmap(telch.corr()) # cmap = 'YlGnBu'
```

Out[315]: <matplotlib.axes._subplots.AxesSubplot at 0x1a28e20780>

