



# 전설이 아닌 현실의 유니콘

Team 2  
PRESENTATION

팀원:  
김태영,  
김나영,  
박근웅,  
조아라,  
정우주,  
오병훈



# INDEX

## Why?

- 원티드가 크레딧잡을 인수한 이유
- 현 시점에서 원티드에게 유니콘은 중요한 이유

## What?

- 대기업과 이미 성장한 상장을 기다리는 유니콘
- 우리가 생각하는 유니콘 기업

## How?

- 고객의 니즈에 맞춘 가설 설정
- 데이터의 기간, 매출액, 월급, 직원수의 증가율의 중요성

## Unicorn



# ABOUT WANTED

왜 크레딧잡을 인수했을까?

1. 비지니스모델
2. 기업 DB를 활용
3. AI를 활용한 채용분야 혁신





# What we do

## 우리가 생각하는 유니콘 기업

부제: 원티드가 유니콘 기업을 찾고 싶은 이유

대기업 or 이미 성장한  
유니콘 기업은 제외!

- 공채 및 직접 채용

매출액 증가율이  
높아야 한다!

- 가고 싶은 회사는 매출이  
잘 나오는 회사.

직원 수, 월급이  
증가하여야 한다!

- 성장하는 기업은 끊임없이  
고용을 할 것이다.





# 유니콘기업 조건 3가지

매출 성장률

연평균 20%

성장성 3개년 평균  
매출 성장률  
20%

직원 수 상승률

반기 13%

현재유니콘 기업들  
지명되기 전,  
반기별 증감율  
중위 값  
13%

월급 증가율

연 4%

일반기업이 받는  
최대 월급 증가율  
4%

>>

# 1. 매출 성장률

```
# 매출표 분석 3년_매출_성장률

y, x = 매출표.shape
for i in range(y):
    flags = 매출표.loc[i, ['to_2016', 'to_2017', 'to_2018', 'to_2019']].notnull().tolist()
    if sum(flags[2:])==2:
        매출표.loc[i, '마지막_연도_매출액'] = 매출표.loc[i, 2019]
        매출표.loc[i, '마지막_연도'] = 2019
        매출표.loc[i, '3년_매출_성장률'] = round(매출표.loc[i, ['to_2017', 'to_2018', 'to_2019']].mean())
        매출표.loc[i, '마지막_증가액'] = 매출표.loc[i, 2019] - 매출표.loc[i, 2018]

    elif sum(flags[1:-1])==2:
        매출표.loc[i, '마지막_연도_매출액'] = 매출표.loc[i, 2018]
        매출표.loc[i, '마지막_연도'] = 2018
        매출표.loc[i, '3년_매출_성장률'] = round(매출표.loc[i, ['to_2016', 'to_2017', 'to_2018']].mean())
        매출표.loc[i, '마지막_증가액'] = 매출표.loc[i, 2018] - 매출표.loc[i, 2017]

    elif sum(flags[:2])==2:
        매출표.loc[i, '마지막_연도_매출액'] = 매출표.loc[i, 2017]
        매출표.loc[i, '마지막_연도'] = 2017
        매출표.loc[i, '3년_매출_성장률'] = round(매출표.loc[i, ['to_2015', 'to_2016', 'to_2017']].mean())
        매출표.loc[i, '마지막_증가액'] = 매출표.loc[i, 2017] - 매출표.loc[i, 2016]

매출표 = 매출표.dropna(subset=['마지막_연도'])
매출표
```

매출표 = 매출표[ (매출표['3년\_매출\_성장률'] >= 40) & (매출표['마지막\_증가액'] >=0) ]



## 2. 직원 수의 성장률

```
# 평균 직원 수
상반기 = [1, 2, 3, 4, 5, 6]
하반기 = [7, 8, 9, 10, 11, 12]
평균_직원표 = pd.DataFrame(columns = ['회사ID']+ [f'{i}_{s}_반기' for i in range(2015, 2020) for s in ['상', '하']])
for co_id in candi:
    data = {f'{i}_{s}_반기': np.nan for i in range(2015, 2020) for s in ['상', '하']}
    data['회사ID'] = co_id
    tmp_df = df_dict[co_id]
    tmp_gp = tmp_df.groupby(['년도'])
    for year, df_year in tmp_gp:
        상반기_조건 = (tmp_df['년도'] == year) & (tmp_df['월'].isin(상반기))
        하반기_조건 = (tmp_df['년도'] == year) & (tmp_df['월'].isin(하반기))
        data[f'{year}_상반기'] = np.round(tmp_df.loc[상반기_조건, '월별_직원수'].mean())
        data[f'{year}_하반기'] = np.round(tmp_df.loc[하반기_조건, '월별_직원수'].mean())
    평균_직원표 = 평균_직원표.append(data, ignore_index=True)
```

평균\_직원표

```
평균_직원표['2015to2016'] = 평균_직원표['2015_하반기']
평균_직원표['2016상to2016하'] = (평균_직원표['2016_하반기'] - 평균_직원표['2016_상반기'])/평균_직원표['2016_상반기']*100
평균_직원표['2016to2017'] = (평균_직원표['2017_상반기'] - 평균_직원표['2016_하반기'])/평균_직원표['2016_하반기']*100
평균_직원표['2017상to2017하'] = (평균_직원표['2017_하반기'] - 평균_직원표['2017_상반기'])/평균_직원표['2017_상반기']*100
평균_직원표['2017to2018'] = (평균_직원표['2018_상반기'] - 평균_직원표['2017_하반기'])/평균_직원표['2017_하반기']*100
평균_직원표['2018상to2018하'] = (평균_직원표['2018_하반기'] - 평균_직원표['2018_상반기'])/평균_직원표['2018_상반기']*100
평균_직원표['2018to2019'] = (평균_직원표['2019_상반기'] - 평균_직원표['2018_하반기'])/평균_직원표['2018_하반기']*100
평균_직원표['2019상to2019하'] = (평균_직원표['2019_하반기'] - 평균_직원표['2019_상반기'])/평균_직원표['2019_상반기']*100
평균_직원표
```

# 직원성장률

```
직원기준 = 평균_직원표.loc[(평균_직원표['2015-2018_평균']>=13, '회사ID')].tolist()
```

### 3. 월급 성장률

```
#평균_월급표
기준1 = ['2015-11', '2015-12', '2016-01', '2016-02', '2016-03', '2016-04', '2016-05', '2016-06']
기준2 = ['2016-07', '2016-08', '2016-09', '2016-10', '2016-11', '2016-12', '2017-01', '2017-02', '2017-03', '2017-04', '2017-05', '2017-06']
기준3 = ['2017-07', '2017-08', '2017-09', '2017-10', '2017-11', '2017-12', '2018-01', '2018-02', '2018-03', '2018-04', '2018-05', '2018-06', '2018-07']
기준4 = ['2018-08', '2018-09', '2018-10', '2018-11', '2018-12', '2019-01', '2019-02', '2019-03', '2019-04', '2019-05', '2019-06', '2019-07']
기준5 = ['2019-08', '2019-09', '2019-10', '2019-11', '2019-12']

평균_월급표 = pd.DataFrame(columns = ['회사ID']+ [f'기준{i}' for i in range(1, 6)])
for co_id in candi:
    data = {f'기준{i}':np.nan for i in range(1, 6)}
    data['회사ID'] = co_id
    tmp_df = df_dict[co_id]
    cri1 = tmp_df['연월'].isin(기준1)
    cri2 = tmp_df['연월'].isin(기준2)
    cri3 = tmp_df['연월'].isin(기준3)
    cri4 = tmp_df['연월'].isin(기준4)
    cri5 = tmp_df['연월'].isin(기준5)

    data['기준1'] = np.round(tmp_df.loc[cri1, '평균월급'].mean())
    data['기준2'] = np.round(tmp_df.loc[cri2, '평균월급'].mean())
    data['기준3'] = np.round(tmp_df.loc[cri3, '평균월급'].mean())
    data['기준4'] = np.round(tmp_df.loc[cri4, '평균월급'].mean())
    data['기준5'] = np.round(tmp_df.loc[cri5, '평균월급'].mean())

평균_월급표 = 평균_월급표.append(data, ignore_index=True)
```

평균\_월급표

```
평균_월급표['1to2'] = (평균_월급표['기준2']-평균_월급표['기준1']) * 100
평균_월급표['2to3'] = (평균_월급표['기준3']-평균_월급표['기준2']) * 100
평균_월급표['3to4'] = (평균_월급표['기준4']-평균_월급표['기준3']) * 100
평균_월급표['4to5'] = (평균_월급표['기준5']-평균_월급표['기준4']) * 100
평균_월급표['평균'] = 평균_월급표[['1to2', '2to3', '3to4', '4to5']].mean(axis=1)
```

# 월급 성장률 4

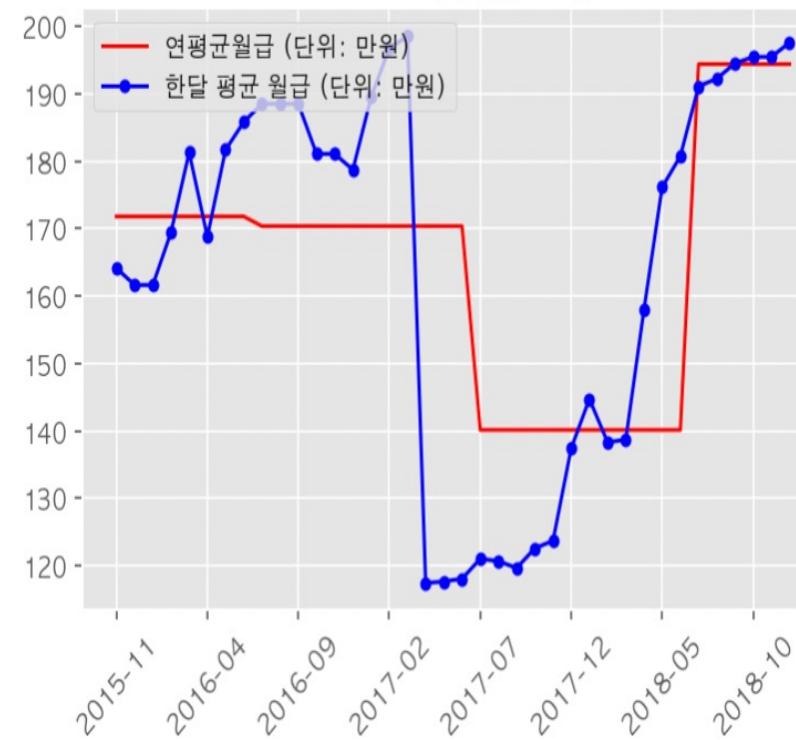
월급기준 = 평균\_월급표.loc[평균\_월급표['평균']>=4, '회사ID'].tolist()

# 회사 ID: 129633

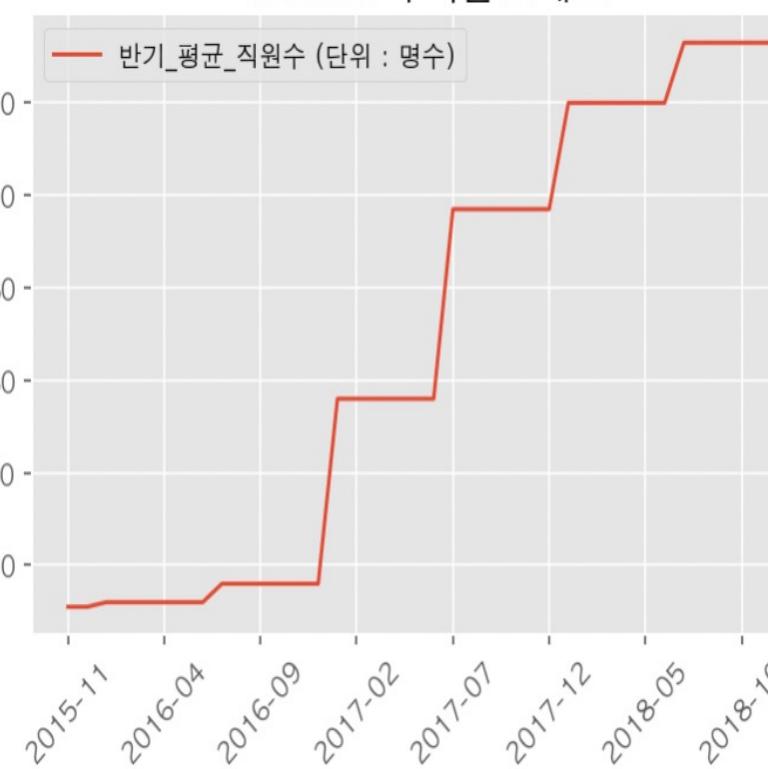
## OUR Unicorn



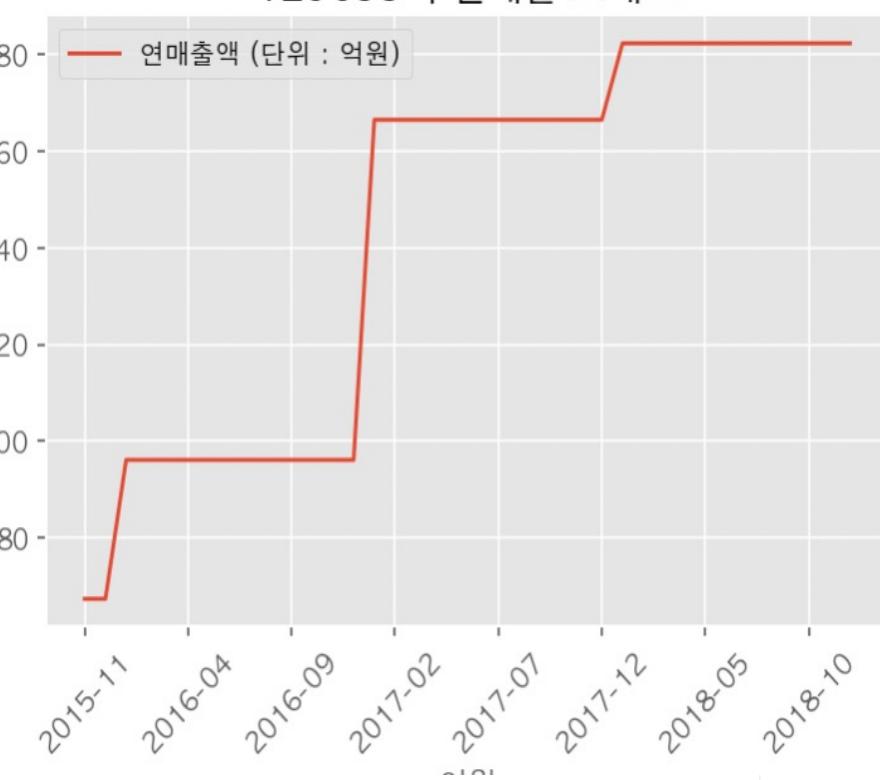
### 129633의 월급 그래프



### 129633의 직원 그래프



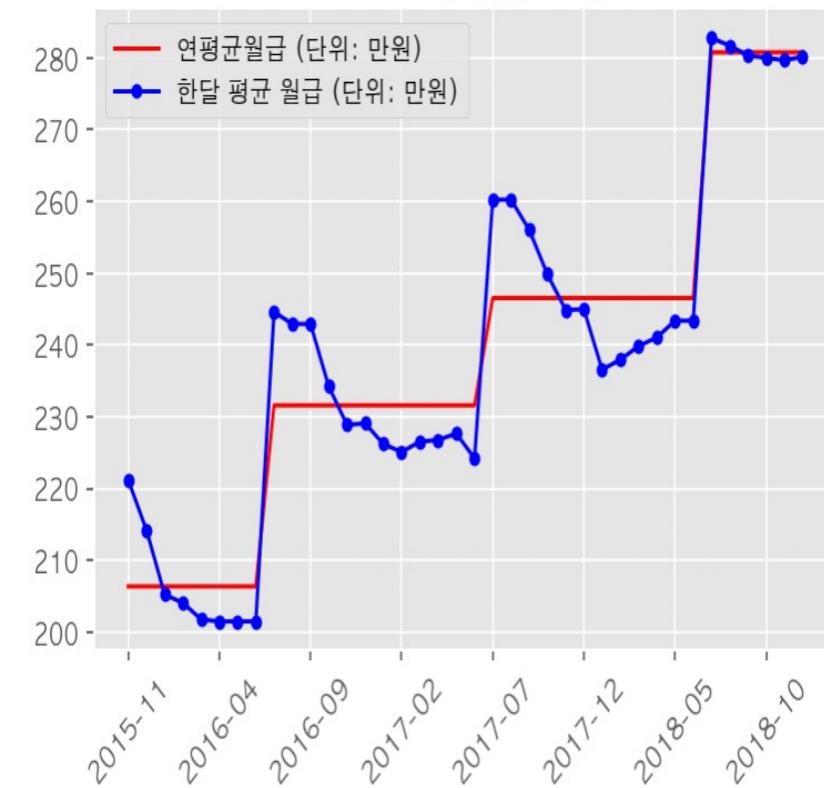
### 129633의 연매출 그래프



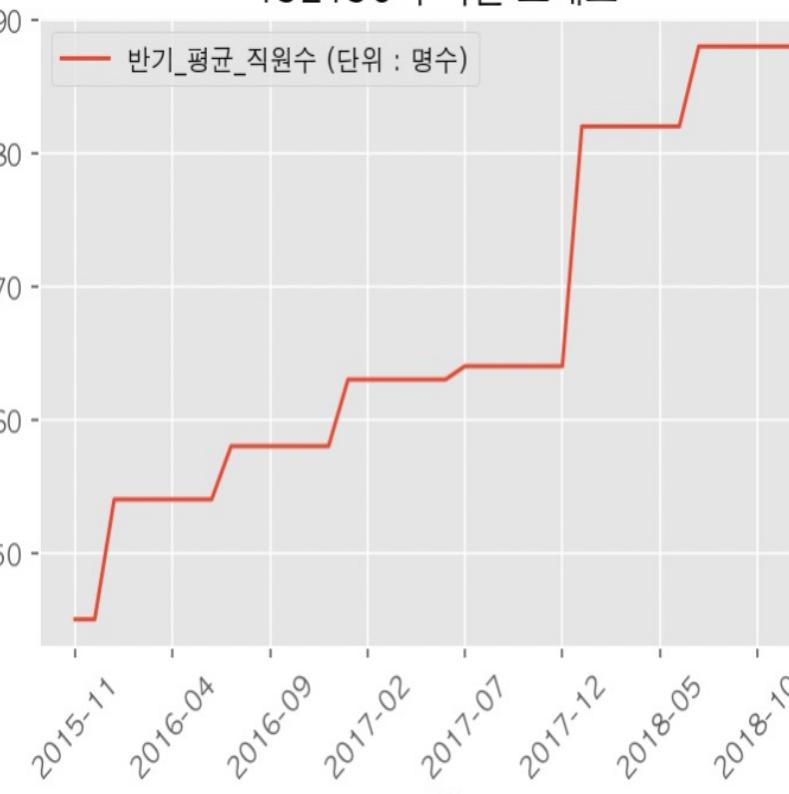
# 회사 ID: 132156

## OUR Unicorn

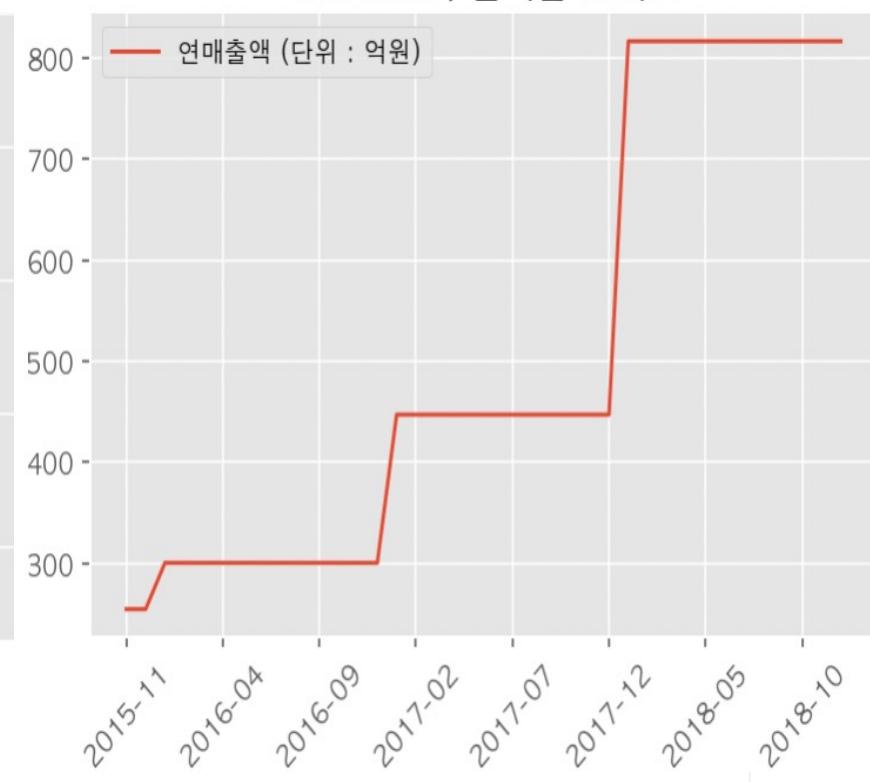
132156의 월급 그래프



132156의 직원 그래프



132156의 연매출 그래프

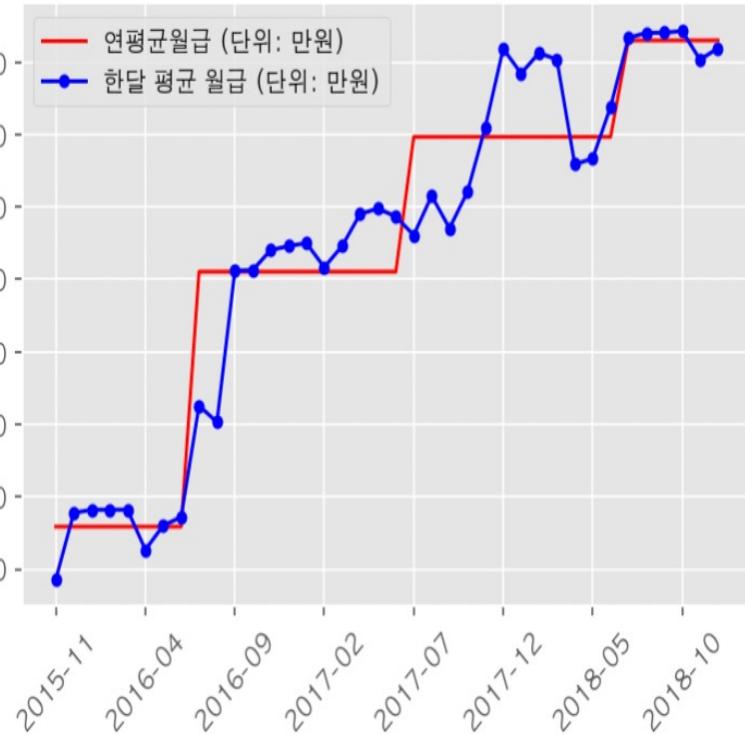


# 회사 ID: 135559

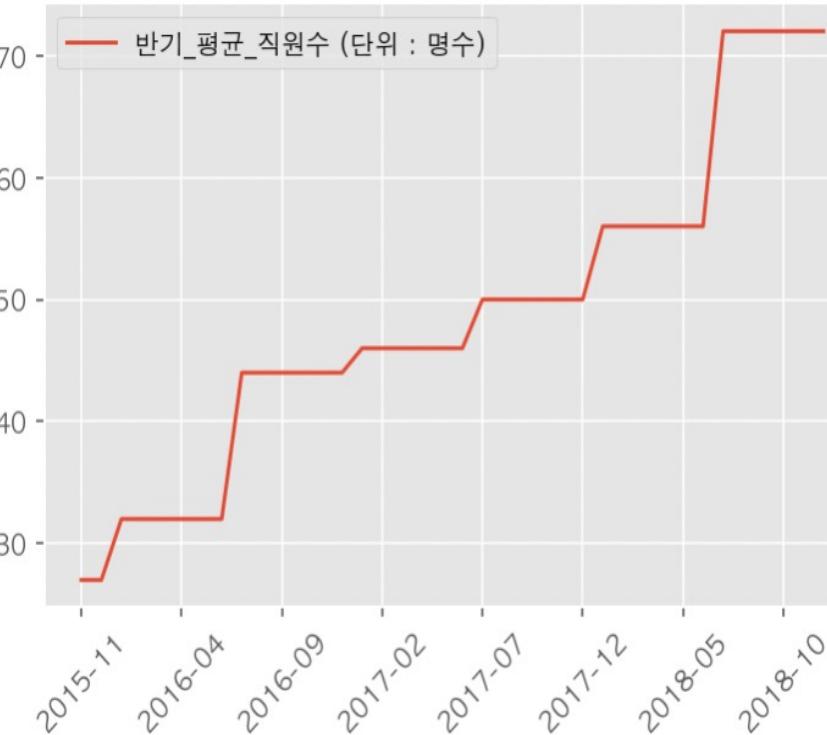
## OUR Unicorn



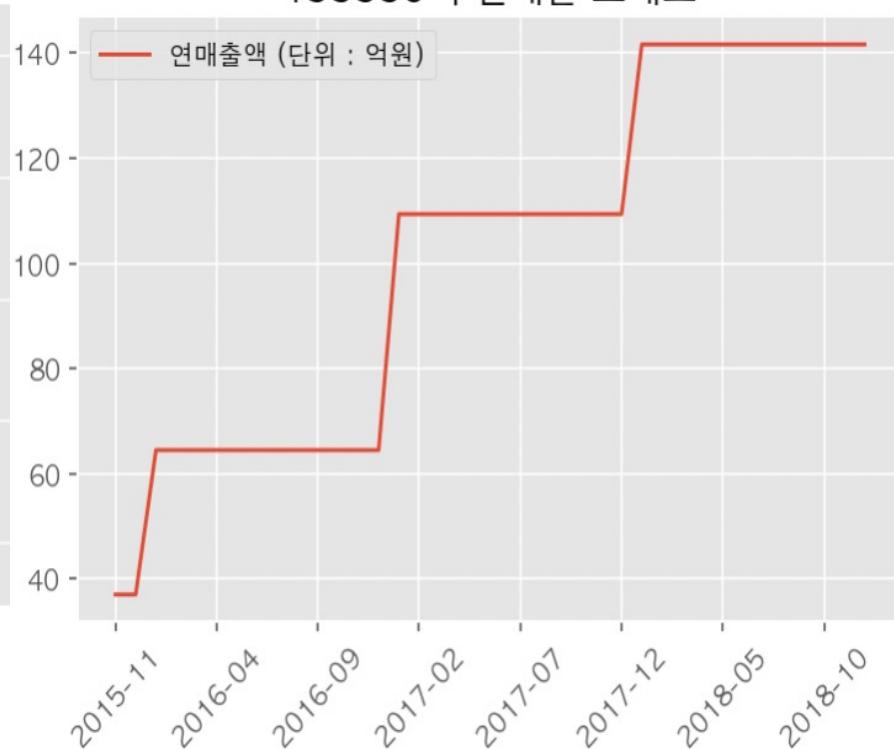
135559의 월급 그래프



135559의 직원 그래프



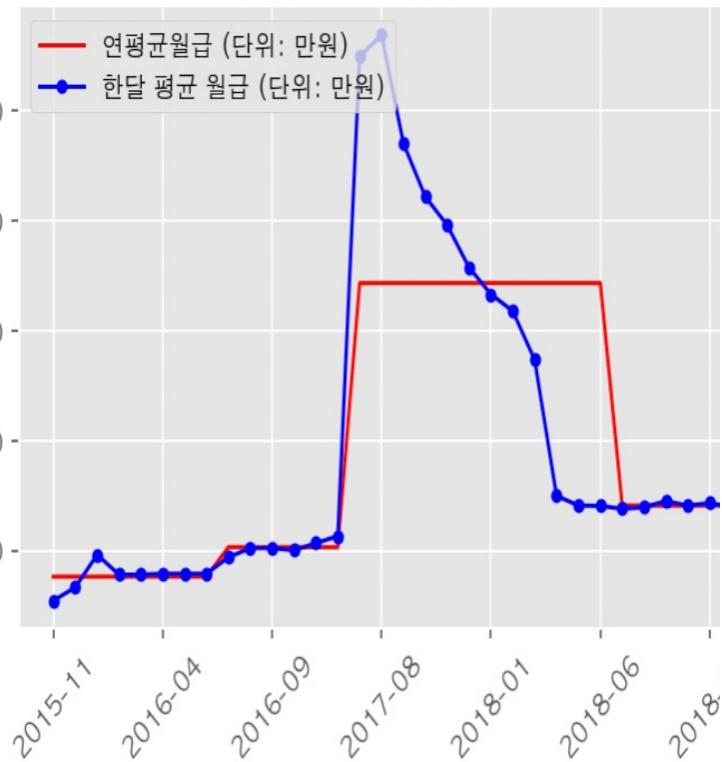
135559의 연매출 그래프



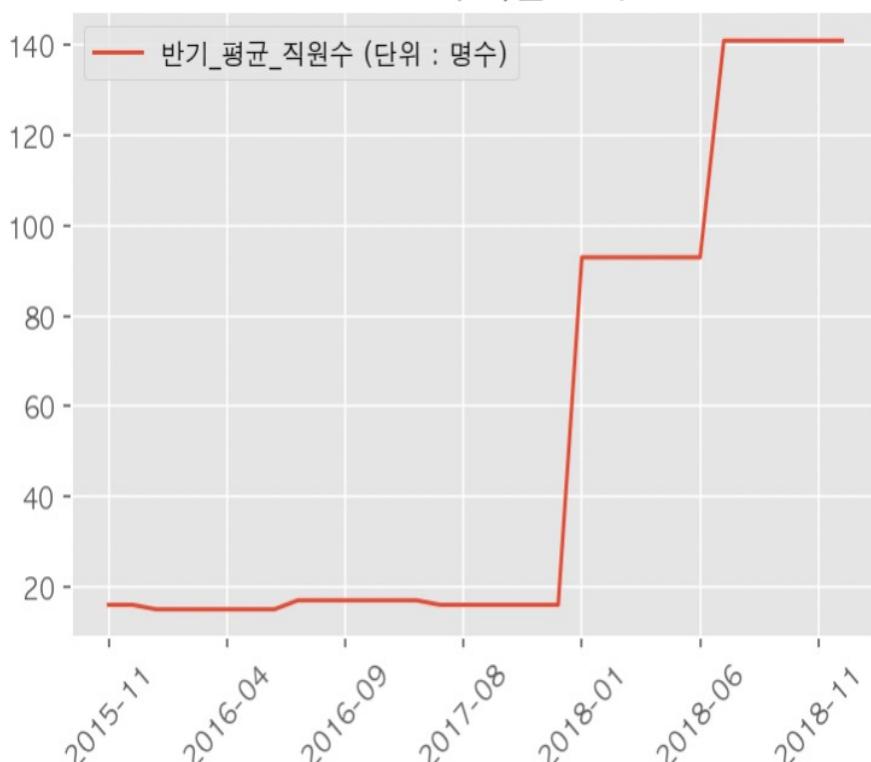
# 회사 ID: 394749

## OUR Unicorn

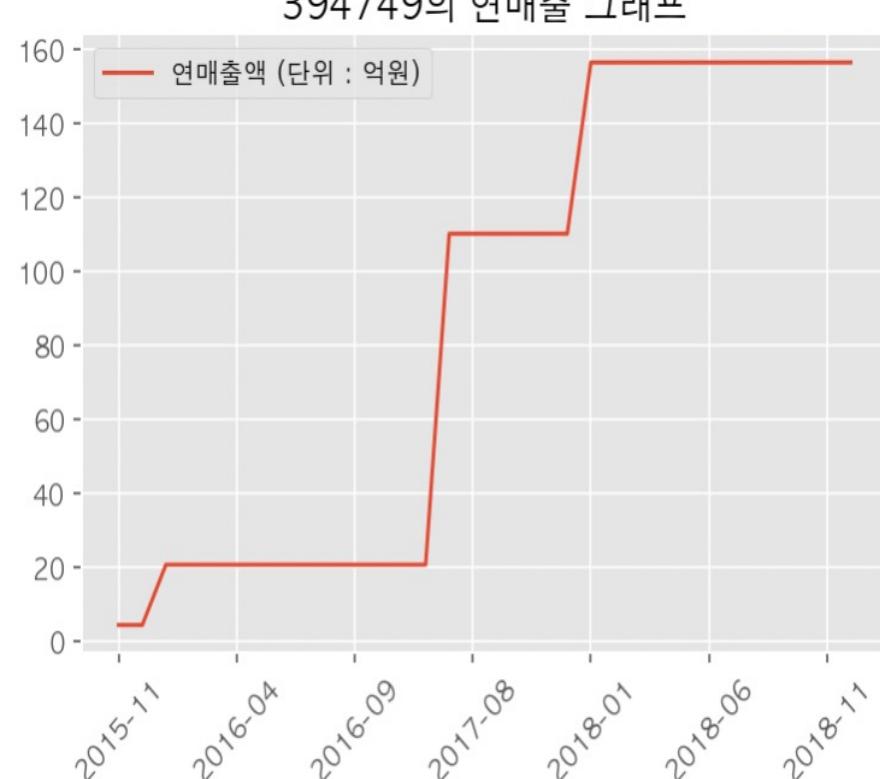
### 394749의 월급 그래프



### 394749의 직원 그래프



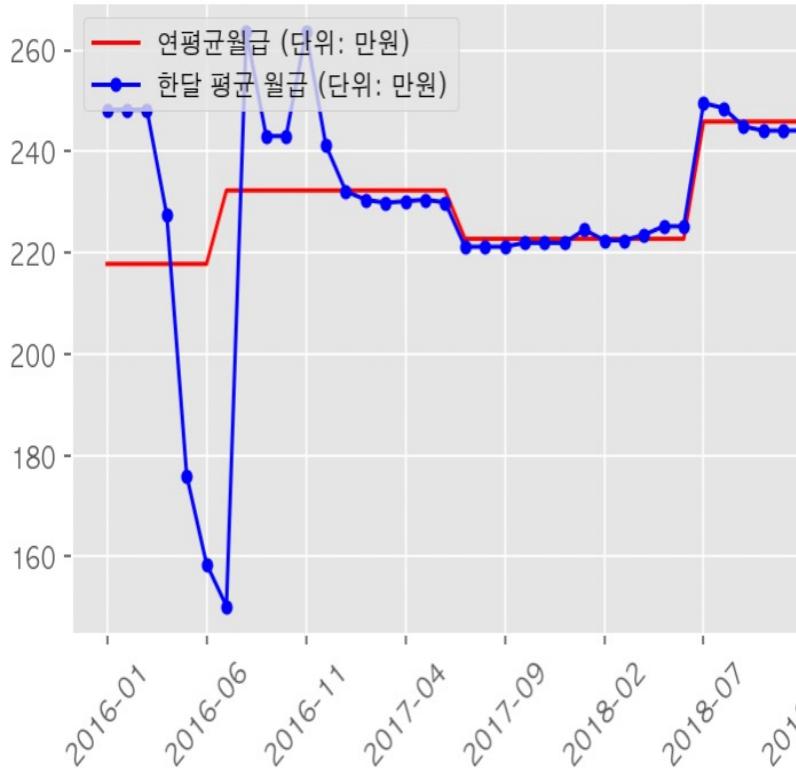
### 394749의 연매출 그래프



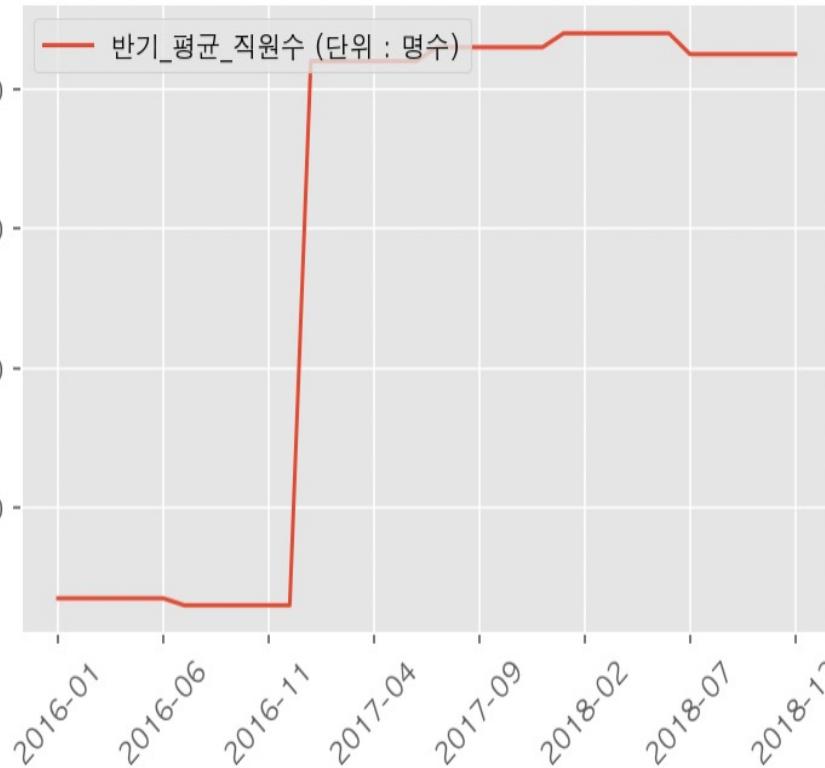
# 회사 ID: 470994

## OUR Unicorn

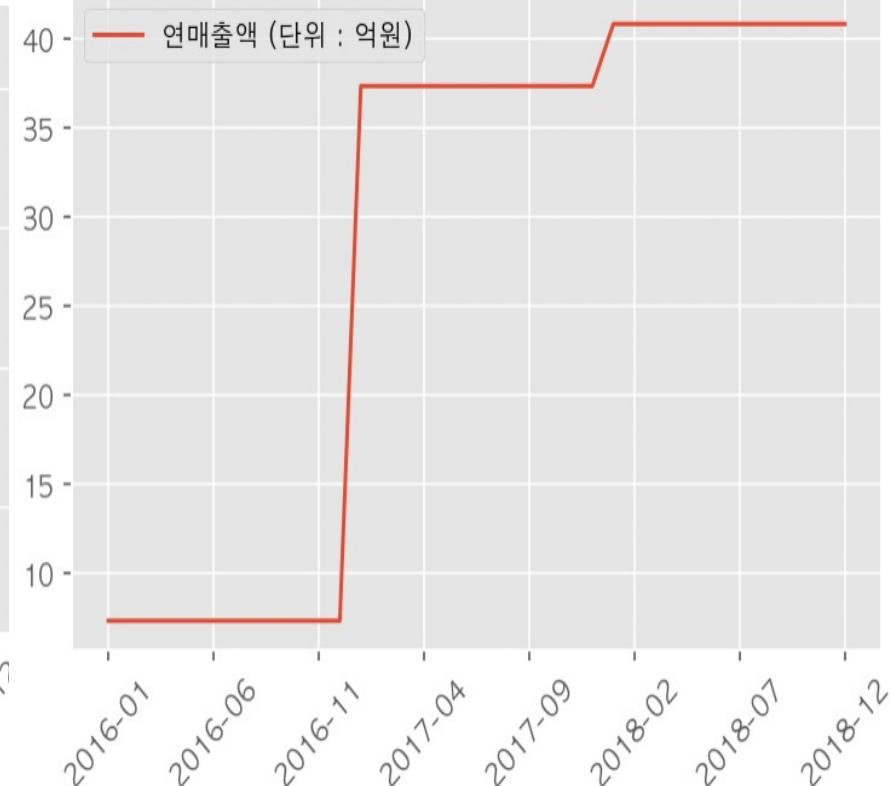
### 470994의 월급 그래프



### 470994의 직원 그래프



### 470994의 연매출 그래프





Question?





Thank you !