

지구온난화 기후변화 예측

홍기준 2017108275

2021.11.24

목차

01 선정 배경

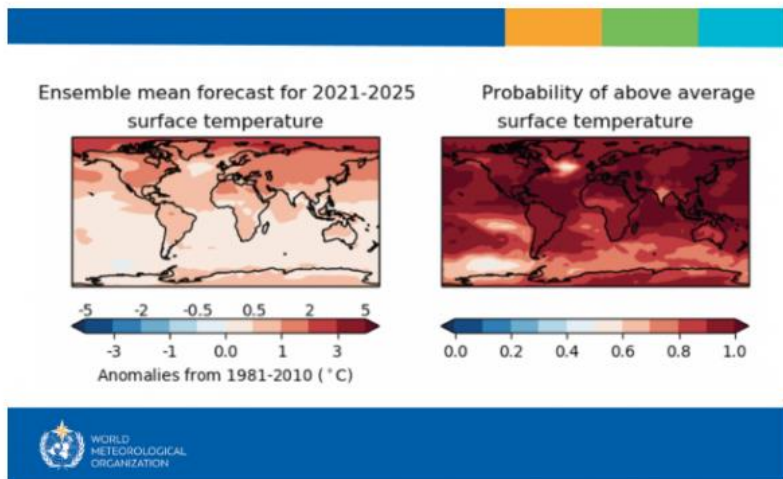
02 데이터 소개

03 코드 분석

04 알고리즘 소개

05 참고자료

선정배경



New climate predictions increase likelihood of temporarily reaching 1.5 °C in next 5 years

Tags: [Climate](#)

27

Published 27 May 2021






Press Release Number: 27052021

다양한 주제의 데이터를 찾아보면서 데이터 분석에 어울리는 주제라고 생각했습니다.
또한 지구온난화에 대한 관심이 많은 만큼 양질의 자료가 많을 것이라고 생각했습니다.

데이터 소개

< Environment_Temperature_change_E_All_Data_NOFLAG.csv (6.26 MB)

Detail Compact Column 10 of 66 columns

# Area Code	▲ Area	# Months Code	▲ Months	# Element Code	▲ Element	▲ Unit	# Y1961	# Y1962
The numerical code of area column, type of area code is an integer.	Countries and Territories (in 2019: 190 countries and 37 other territorial entities.), type of area is an object.	The numerical code of months column, type of months code is an integer.	Months, Seasons, Meteorological year, type of months is an object.	The numerical code of element column, type of element code is an integer.	'Temperature change', 'Standard Deviation', type of element is an object.	Celsius degrees °C, type of unit is an object.	1961	1962
	284 unique values		17 unique values		2 unique values	1 unique value		
117	Republic of Korea	7010	October	7271	Temperature change	°C	1.510000	-0.494000
117	Republic of Korea	7010	October	6078	Standard Deviation	°C	0.964000	0.964000
117	Republic of Korea	7011	November	7271	Temperature change	°C	1.687000	-0.529000
117	Republic of Korea	7011	November	6078	Standard Deviation	°C	1.253000	1.253000
117	Republic of Korea	7012	December	7271	Temperature change	°C	0.185000	1.496000
117	Republic of Korea	7012	December	6078	Standard Deviation	°C	1.788000	1.788000
117	Republic of Korea	7016	Dec-Jan-Feb	7271	Temperature change	°C	-0.737000	0.320000
117	Republic of Korea	7016	Dec-Jan-Feb	6078	Standard Deviation	°C	1.108000	1.108000
117	Republic of Korea	7017	Mar-Apr-May	7271	Temperature change	°C	0.613000	-0.312000
117	Republic of Korea	7017	Mar-Apr-May	6078	Standard Deviation	°C	0.538000	0.538000
117	Republic of Korea	7018	Jun-Jul-Aug	7271	Temperature change	°C	1.201000	0.022000
117	Republic of Korea	7018	Jun-Jul-Aug	6078	Standard Deviation	°C	0.725000	0.725000
117	Republic of Korea	7019	Sep-Oct-Nov	7271	Temperature change	°C	1.499000	-0.383000
117	Republic of Korea	7019	Sep-Oct-Nov	6078	Standard Deviation	°C	0.700000	0.700000
117	Republic of Korea	7020	Meteorological year	7271	Temperature change	°C	0.644000	-0.088000
117	Republic of Korea	7020	Meteorological year	6078	Standard Deviation	°C	0.504000	0.504000

미국항공우주국에서 배포한 지구 표면 온도 변화 데이터를 기반으로 FAOST 온도 변화 도메인에서 국가별 평균 표면 온도 변화 통계를 배포

시간 범위: 1961-2019

주기: 월별, 계절별, 연도별

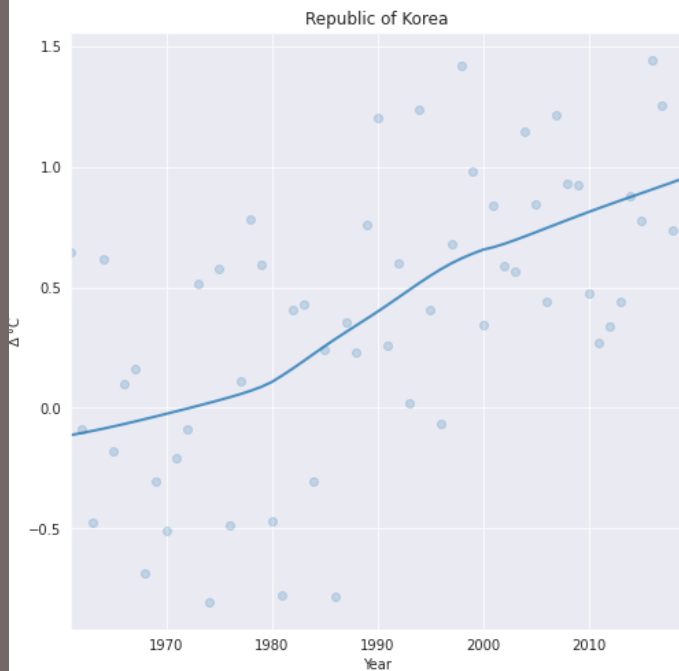
기준기간: 1951-1980

측정 단위: 섭씨 °C

참조 기간: 월, 계절, 기상 연도

데이터 소개

```
Korea = country_df(df, 'Republic of Korea')  
country_regplot(Korea, 'Meteorological year')
```



함수에 원하는 나라를 입력하고 변수에 저장

한국은 Republic of Korea로 저장

연도별 기온변화를 그래프에 산점도로 표현

기온 변화의 정도를 선으로 표현

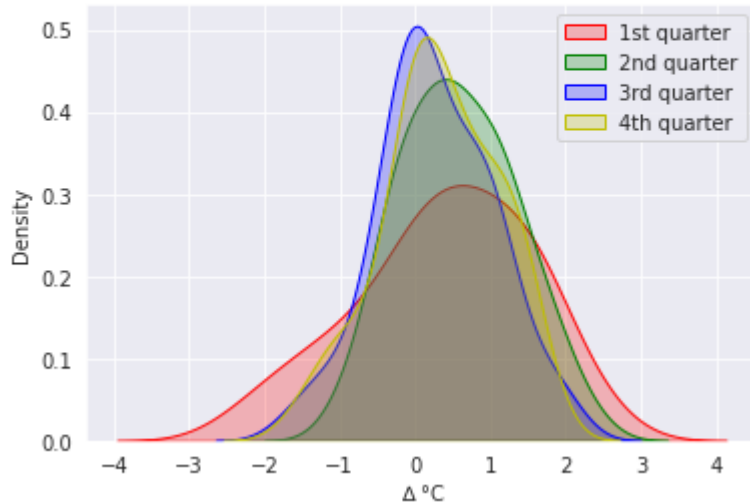
X - 연도

Y - 기온 변화

regplot

데이터 소개

```
country_densityplot(Korea)
```



densityplot

계절별 온도 변화의 분포를 밀도 그래프로 표현

1년을 4개의 계절로 분류(1st- 4th quarter)

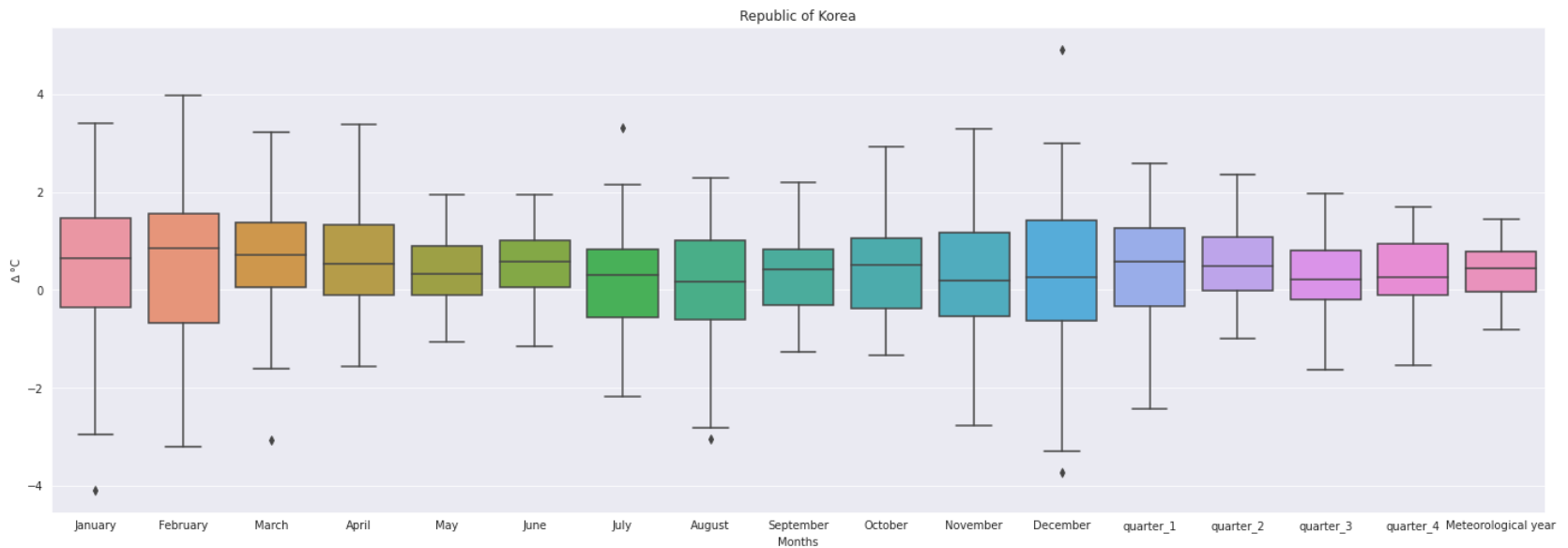
계절 마다 기온 변화 의 밀도를 그래프로 출력

x - 기온 변화

Y - 분포 정도(밀도)

데이터 소개

```
country_boxplot(months_df(df, 'Republic of Korea'), 'Republic of Korea')
```



boxplot

월별, 계절 별 기온 변화의 분포를 한눈에 볼 수 있도록 시각화

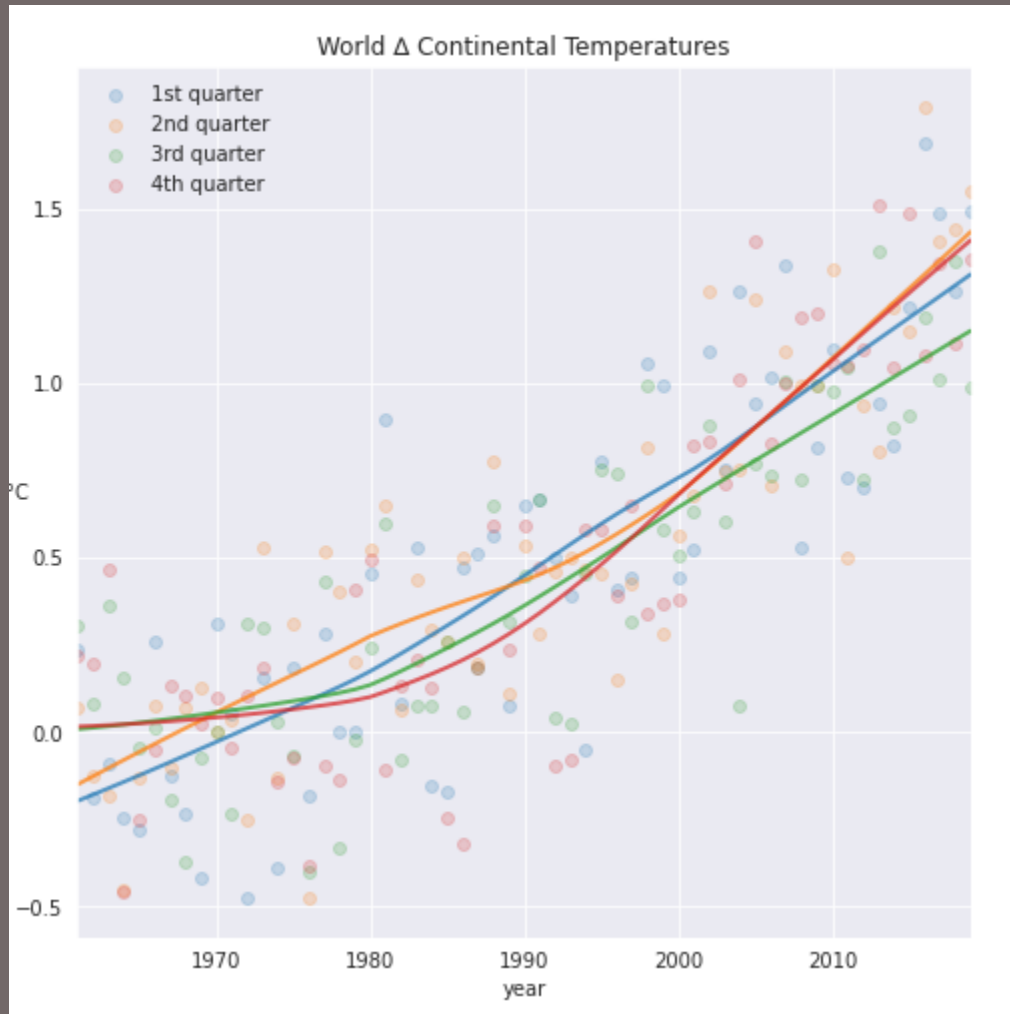
x - 월, 계절, 표준기상년도

y - 기온 변화

Meteorological year = 표준기상년도

1년 동안 일사량과 기상 요소의 시간별 값을 나열

데이터 소개



전세계 국가, 지역의 기온을 종합해서

regplot으로 표현

x - 연도

y - 기온 변화

코드 분석

```
germany = country_df(df, 'Germany')  
country_regplot(germany, 'Meteorological year')
```

독일의 국가 데이터 호출
한국은 Republic of Korea

여러 국가의 기온이 저장된 csv파일에서 원하는 나라를 지정하고
regplot으로 표현하는 코드

```
country_densityplot(germany)
```

지정된 국가의 기온 정보를 densityplot으로 표현하는 코드

```
country_boxplot(months_df(df, 'Germany'), 'Germany')
```

지정된 국가의 기온 정보를 boxplot으로 표현하는 코드

코드 분석

```
world = sum(continent / 7
world.index.name = 'world'
world.head()
```

7개 대륙의 기온 데이터

(북아메리카, 남아메리카, 남, 아프리카, 유럽, 아시아, 오세아니아)

7개 대륙(continent)의 기온 데이터를 합하고 7로 나눈 평균값을 world에 저장

```
def world_regplot(data):
    labels = ['1st quarter', '2nd quarter', '3rd quarter', '4th quarter']
    plt.figure(figsize=(8,8))

    for i, q in enumerate(quarters):
        sns.regplot(data=data, x='year', y=q, fit_reg=True, lowess=True, label=labels[i], scatter_kws={'alpha':0.2}, line_kws={'lw':2, 'alpha':0.75})

    plt.gca().set_ylabel('Δ °C', rotation=0)
    plt.gca().set_title('World Δ Continental Temperatures')
    plt.legend(loc='best', frameon=False)
    plt.show()

world_regplot(world)
```

그래프의 라벨을 설정(계절 별)

전세계의 기온을 시각화하는 함수 world_regplot 정의

알고리즘 소개

```
from sklearn.linear_model import Lasso
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import AdaBoostRegressor
```

Lasso, DecisionTreeRegressor, AdaBoostRegressor 모델을 사용

```
ntrain = train.shape[0]
y_train = train["observedMaxTemp"]
y_test = test["observedMaxTemp"]
train = train[train.columns[:-1]]
train = train.append(test)
del train["observedMaxTemp"]
train.shape
```

학습용 데이터와 시험용 데이터를 등록

```
x_train_st, x_test_st, y_train_st, y_test_st = train_test_split(x_train_np, y_train_np, train_size=0.7, random_state=1)
```

학습용 데이터와 시험용 데이터를 설정한대로 분배

알고리즘 소개

```
def cv_score(model):  
  
    cv = GridSearchCV(estimator=model  
                      ,param_grid=param_dict  
                      ,n_jobs=4  
                      ,scoring=rmse_scorer #"neg_mean_squared_error"  
                      ,cv = 5 #[x_train_np,y_train_np]  
                      ,return_train_score=True  
                      ,verbose=1)  
    bst_model = cv.fit(x_train_st,y_train_st).best_estimator_ → 분배한 데이터로 학습을 진행  
  
    tr_cv = np.mean(np.sqrt(-cross_val_score(bst_model,x_train_st,y_train_st,scoring="neg_mean_squared_error", cv = 5, n_jobs=4, verbose=1)))  
    ts_cv = np.mean(np.sqrt(-cross_val_score(bst_model,x_test_st,y_test_st,scoring="neg_mean_squared_error", cv = 5, n_jobs=4, verbose=1)))  
    kag_tr_cv = np.mean(np.sqrt(-cross_val_score(bst_model,x_train_np,y_train_np,scoring="neg_mean_squared_error", cv = 5, n_jobs=4, verbose=1)))  
  
    # kag_ts_cv = np.mean(np.sqrt(-cross_val_score(bst_model,x_train_np,y_train_np,scoring="neg_mean_squared_error", cv = 5, n_jobs=4, verbose=1)))  
  
    print(bst_model)  
    print(tr_cv,ts_cv,kag_tr_cv) → 정확도가 높은 모델과 데이터 출력  
  
    return [cv, tr_cv, ts_cv, kag_tr_cv]
```

머신러닝 모델의 정확도를 계산하는 함수

fit을 통해 테스트를 진행하고 정확도가 가장 높은 모델을 출력하고 학습, 시험 데이터를 출력

참고자료

전세계의 기온 변화 데이터

<https://www.kaggle.com/sevgisarak/temperature-change>

기온 변화 예측 알고리즘

<https://www.kaggle.com/mineshjethva/weather-forecasting-lasso-gridcv>

인공지능 Github

<https://github.com/yungbyun/ai>

감사합니다.