In [ ]:

```python

```

In [2]:

```python
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as plt
```

In [3]:

```python
df = pd.read_excel('1553768847_housing.xlsx')
```

In [77]:

```python
df.head()
```

Out[77]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	ocean_proximity	median_hou
0	-122.23	37.88	41	880	129.0	322	126	8.3252	NEAR BAY	
1	-122.22	37.86	21	7099	1106.0	2401	1138	8.3014	NEAR BAY	
2	-122.24	37.85	52	1467	190.0	496	177	7.2574	NEAR BAY	
3	-122.25	37.85	52	1274	235.0	558	219	5.6431	NEAR BAY	
4	-122.25	37.85	52	1627	280.0	565	259	3.8462	NEAR BAY	

In [11]:

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   longitude           20640 non-null  float64
 1   latitude            20640 non-null  float64
 2   housing_median_age  20640 non-null  int64  
 3   total_rooms         20640 non-null  int64  
 4   total_bedrooms      20433 non-null  float64
 5   population          20640 non-null  int64  
 6   households          20640 non-null  int64  
 7   median_income       20640 non-null  float64
 8   ocean_proximity     20640 non-null  object 
 9   median_house_value  20640 non-null  int64  
dtypes: float64(4), int64(5), object(1)
memory usage: 1.6+ MB
```

In [233…

```python
len(df.columns)
```

Out[233… 13

In [278…

```python
Columname=[]
def iter():
    for col in range(len(df.columns)):
        colname= df.columns[col]
        Columname.append(colname)
    return Columname
```

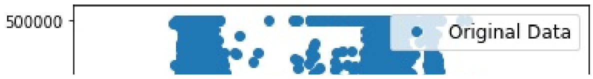
In [342…

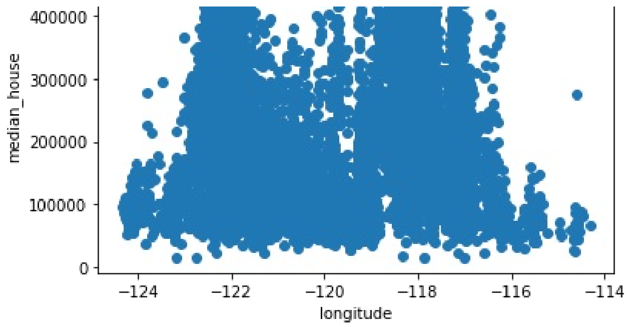
```python
def graphcustomfunc(cols):

    for col in cols:
        fig,axs=plt.subplots(1,1,sharey=True)
        plt.plot(df[col],df['median_house_value'],'o', label='Original Data')
        plt.xlabel(col)
        plt.ylabel('median_house')
        #plt.plot(X_med_inc_test,y_pred1lr,'y-', label='fitted line',lw=4)
        plt.legend(loc=1, fontsize=12)
```

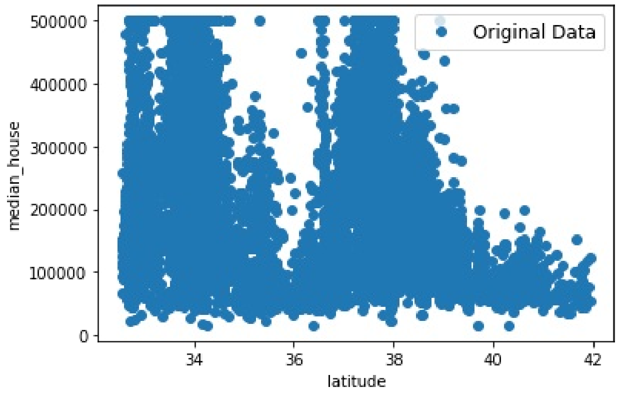
In [343…

```python
#df[Columname].apply(graphcustomfunc)
graphcustomfunc(Columname)
```

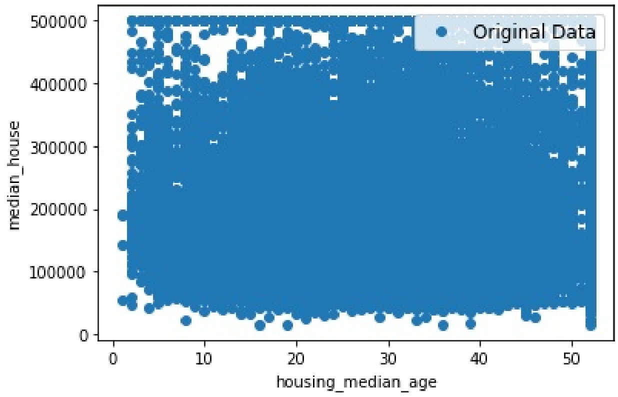




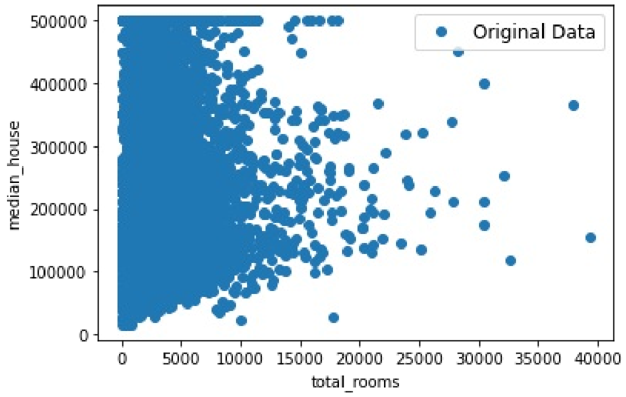
median_house
400000
300000
200000
100000
0
−124
−122
−120
−118
−116
−114
longitude



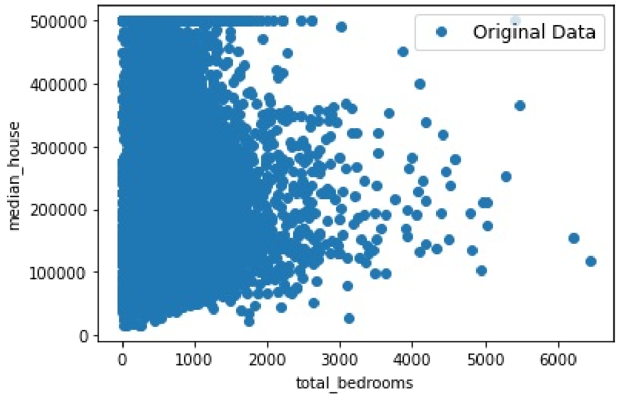
Original Data
median_house
500000
400000
300000
200000
100000
0
34
36
38
40
42
latitude



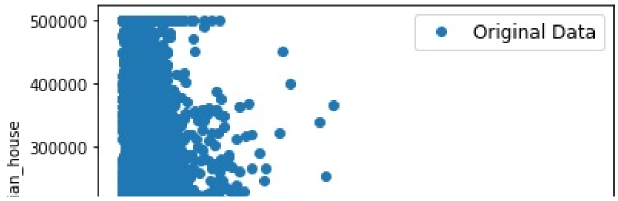
Original Data
median_house
500000
400000
300000
200000
100000
0
0
10
20
30
40
50
housing_median_age



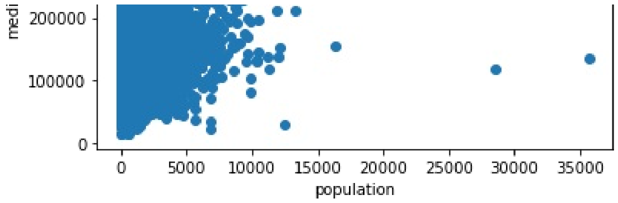
Original Data
median_house
500000
400000
300000
200000
100000
0
0
5000
10000
15000
20000
25000
30000
35000
40000
total_rooms



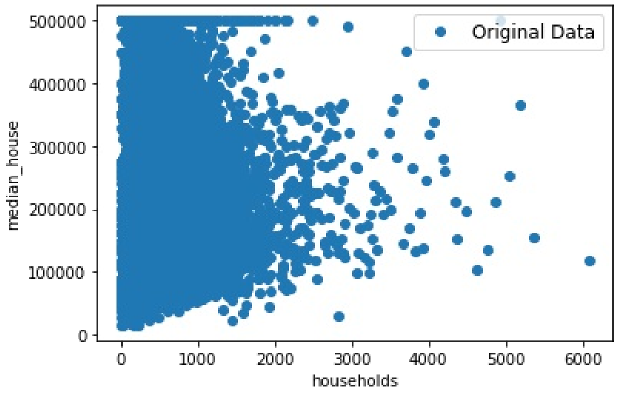
Original Data
median_house
500000
400000
300000
200000
100000
0
0
1000
2000
3000
4000
5000
6000
total_bedrooms



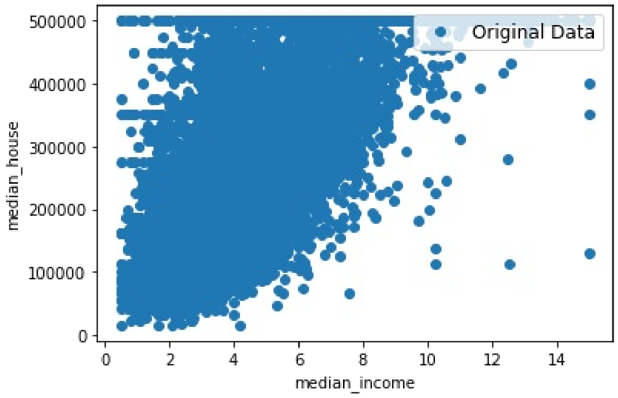
Original Data
ian_house
500000
400000
300000



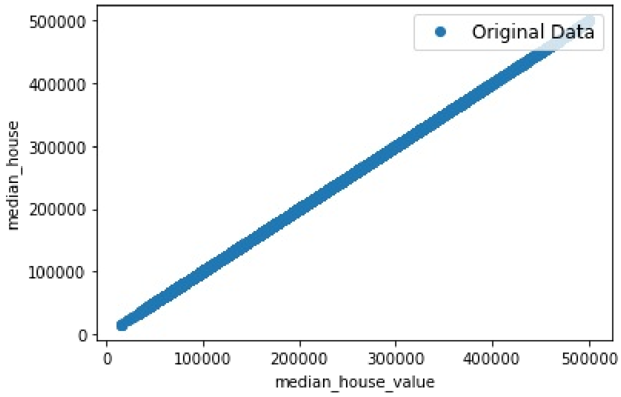
medi
200000
100000
0
0
5000
10000
15000
20000
25000
30000
35000
population



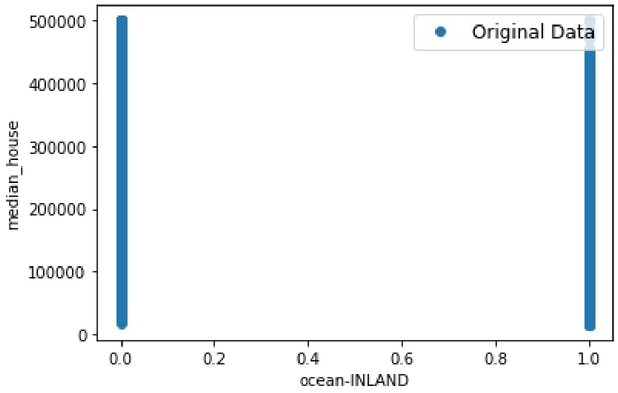
Original Data
median_house
500000
400000
300000
200000
100000
0
0
1000
2000
3000
4000
5000
6000
households



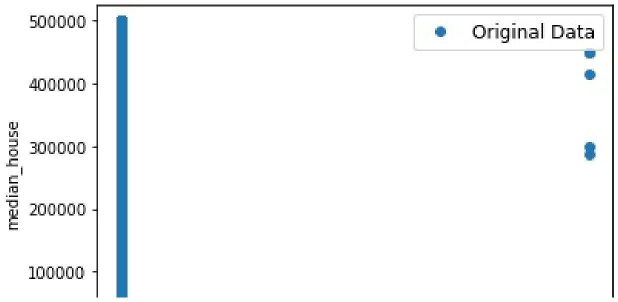
Original Data
median_house
500000
400000
300000
200000
100000
0
0
2
4
6
8
10
12
14
median_income



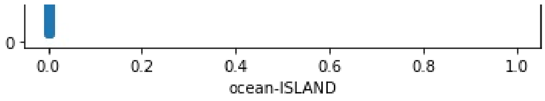
Original Data
median_house
500000
400000
300000
200000
100000
0
100000
200000
300000
400000
500000
median_house_value

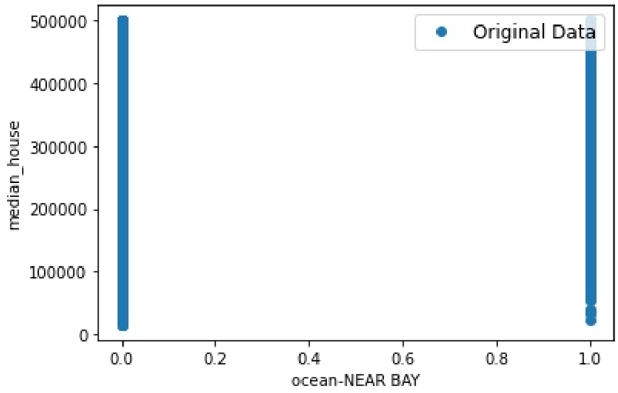


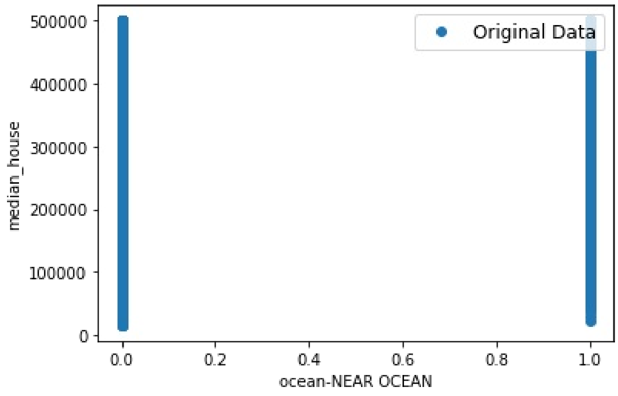
Original Data
median_house
500000
400000
300000
200000
100000
0
0.0
0.2
0.4
0.6
0.8
1.0
ocean-INLAND



Original Data
median_house
500000
400000
300000
200000
100000







In [279…

```
iter()
```

Out[279…

```
['longitude',
 'latitude',
 'housing_median_age',
 'total_rooms',
 'total_bedrooms',
 'population',
 'households',
 'median_income',
 'median_house_value',
 'ocean-INLAND',
 'ocean-ISLAND',
 'ocean-NEAR BAY',
 'ocean-NEAR OCEAN']
```

In [13]:

```
df.describe()
```

Out[13]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_h
count	20640.000000	20640.000000	20640.000000	20640.000000	20433.000000	20640.000000	20640.000000	20640.000000	20
mean	-119.569704	35.631861	28.639486	2635.763081	537.870553	1425.476744	499.539680	3.870671	206
std	2.003532	2.135952	12.585558	2181.615252	421.385070	1132.462122	382.329753	1.899822	115
min	-124.350000	32.540000	1.000000	2.000000	1.000000	3.000000	1.000000	0.499900	14
25%	-121.800000	33.930000	18.000000	1447.750000	296.000000	787.000000	280.000000	2.563400	119
50%	-118.490000	34.260000	29.000000	2127.000000	435.000000	1166.000000	409.000000	3.534800	179
75%	-118.010000	37.710000	37.000000	3148.000000	647.000000	1725.000000	605.000000	4.743250	264
max	-114.310000	41.950000	52.000000	39320.000000	6445.000000	35682.000000	6082.000000	15.000100	500

In [ ]:

```

```

In [16]:

```
df.columns
```

Out[16]:

```
Index(['longitude', 'latitude', 'housing_median_age', 'total_rooms',
       'total_bedrooms', 'population', 'households', 'median_income',
       'ocean_proximity', 'median_house_value'],
      dtype='object')
```

In [43]:

```
df.head()
```

Out[43]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	ocean_proximity	median_hou
0	-122.23	37.88	41	880	129.0	322	126	8.3252	NEAR BAY	
1	-122.22	37.86	21	7099	1106.0	2401	1138	8.3014	NEAR BAY	
2	-122.24	37.85	52	1467	190.0	496	177	7.2574	NEAR BAY	
3	-122.25	37.85	52	1274	235.0	558	219	5.6431	NEAR BAY	
4	-122.25	37.85	52	1627	280.0	565	259	3.8462	NEAR BAY	

In [46]:

```
df.iloc[1]
```

Out[46]:

```
array([-122.22, 37.86, 21, 7099, 1106.0, 2401, 1138, 8.3014, 'NEAR BAY',
       358500], dtype=object)
```

In [48]:

```
df.columns
```

Out[48]:

```
Index(['longitude', 'latitude', 'housing_median_age', 'total_rooms',
       'total_bedrooms', 'population', 'households', 'median_income',
       'ocean_proximity', 'median_house_value'],
      dtype='object')
```

In [57]:

```
df['ocean_proximity'].value_counts()
```

Out[57]:

```
<1H OCEAN     9136
INLAND        6551
NEAR OCEAN    2658
NEAR BAY      2290
ISLAND           5
Name: ocean_proximity, dtype: int64
```

In [4]:

```
df.isnull().sum()
```

Out[4]:

```
longitude               0
latitude                0
housing_median_age      0
total_rooms             0
total_bedrooms        207
population              0
households              0
median_income           0
ocean_proximity         0
median_house_value      0
dtype: int64
```

In [6]:

```
df['total_bedrooms']=df['total_bedrooms'].fillna(df['total_bedrooms'].mean())
df.isnull().sum()
```

Out[6]:

```
longitude             0
latitude              0
housing_median_age    0
```

```
total_rooms           0
total_bedrooms        0
population            0
households            0
median_income         0
ocean_proximity       0
median_house_value    0
dtype: int64
```

Converting Categorical Into Numerical

In [146…

```
#Le=LabelEncoder()
#Le.fit(df.ocean_proximity)
#df['ocean_proximity']=Le.transform(df.ocean_proximity)
```

In [151…

```
ocean=pd.get_dummies(df['ocean_proximity'], drop_first=True, prefix_sep='-', prefix='ocean')
```

In [152…

```
ocean.head()
```

Out[152…

	ocean-INLAND	ocean-ISLAND	ocean-NEAR BAY	ocean-NEAR OCEAN
0	0	0	1	0
1	0	0	1	0
2	0	0	1	0
3	0	0	1	0
4	0	0	1	0

In [153…

```
df=pd.concat([df,ocean],axis=1)
```

In [154…

```
df.drop(['ocean_proximity'], axis=1, inplace=True)
```

In [155…

```
df
```

Out[155…

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	o INI
0	-122.23	37.88	41	880	129.0	322	126	8.3252	452600	
1	-122.22	37.86	21	7099	1106.0	2401	1138	8.3014	358500	
2	-122.24	37.85	52	1467	190.0	496	177	7.2574	352100	
3	-122.25	37.85	52	1274	235.0	558	219	5.6431	341300	
4	-122.25	37.85	52	1627	280.0	565	259	3.8462	342200	
...	...	...	...	...	...	...	...	...	...	
20635	-121.09	39.48	25	1665	374.0	845	330	1.5603	78100	
20636	-121.21	39.49	18	697	150.0	356	114	2.5568	77100	
20637	-121.22	39.43	17	2254	485.0	1007	433	1.7000	92300	
20638	-121.32	39.43	18	1860	409.0	741	349	1.8672	84700	
20639	-121.24	39.37	16	2785	616.0	1387	530	2.3886	89400	

20640 rows × 13 columns

Select Data and Response

In [156…

```
x=df.drop('median_house_value', axis=1)
y=df['median_house_value']
```

In [157…

```
from sklearn.model_selection import train_test_split
X_train, X_test , Y_train ,Y_test =train_test_split(x,y , test_size=0.3, random_state=42)
```

In [158…
```
X_train.shape, X_test.shape , Y_train.shape ,Y_test.shape
```

Out[158…
```
((14448, 12), (6192, 12), (14448,), (6192,))
```

In [159…
```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
X_train_std=sc.fit_transform(X_train)
X_test_std= sc.transform(X_test)
```

In [160…
```
from sklearn.linear_model import LinearRegression
lr= LinearRegression()
lr.fit(X_train_std,Y_train)
```

Out[160…
```
LinearRegression()
```

In [161…
```
y_pred=lr.predict(X_test_std)
```

In [175…
```
from sklearn.metrics import mean_squared_error,r2_score
print(r2_score(Y_test,y_pred))
print(mean_squared_error(Y_test,y_pred))
```

```
0.6395785380523742
4730676245.231668
```

In [ ]:
```

```

In [ ]:
```
7.Bonus Excercise
```

In [191…
```
X_med_inc_train=X_train[['median_income']]
X_med_inc_test=X_test[['median_income']]
```

In [188…
```
type(X_med_inc_train)
```

Out[188…
```
pandas.core.frame.DataFrame
```

In [192…
```
lr1=LinearRegression()
lr1.fit(X_med_inc_train,Y_train)
```

Out[192…
```
LinearRegression()
```

In [194…
```
y_pred1lr=lr1.predict(X_med_inc_test)
```

In [196…
```
print(r2_score(Y_test,y_pred1lr))
print(mean_squared_error(Y_test,y_pred1lr))
```

```
0.4729319258997021
6917979868.048501
```

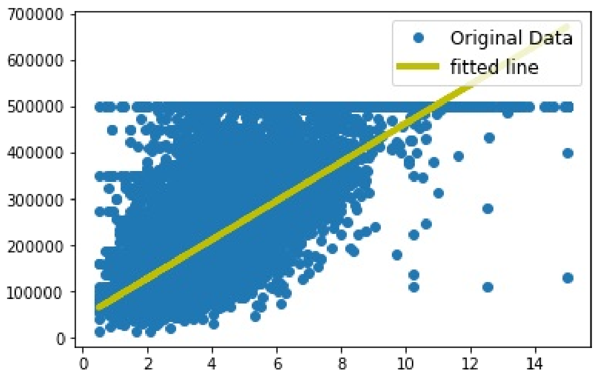
In [195…
```
y_pred1lr
```

Out[195…
```
array([115101.61806807, 150652.22793035, 190330.40536516, ...,
       191664.4418957 , 197435.50901838, 172427.55148675])
```

In [213…
```
import matplotlib.pyplot as plt
```

```
plt.plot(figuresize=(20,20))
plt.plot(X_med_inc_train,Y_train,'o', label='Original Data')
plt.plot(X_med_inc_test,y_pred1lr,'y-', label='fitted line',lw=4)
plt.legend(loc=1, fontsize=12)
```

Out[213… <matplotlib.legend.Legend at 0x2023a088100>



In [58]:

```
x=df.iloc[:,:-1]
x
```

Out[58]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	ocean_proximity
0	-122.23	37.88	41	880	129.0	322	126	8.3252	NEAR BAY
1	-122.22	37.86	21	7099	1106.0	2401	1138	8.3014	NEAR BAY
2	-122.24	37.85	52	1467	190.0	496	177	7.2574	NEAR BAY
3	-122.25	37.85	52	1274	235.0	558	219	5.6431	NEAR BAY
4	-122.25	37.85	52	1627	280.0	565	259	3.8462	NEAR BAY
...	...	...	...	...	...	...	...	...	...
20635	-121.09	39.48	25	1665	374.0	845	330	1.5603	INLAND
20636	-121.21	39.49	18	697	150.0	356	114	2.5568	INLAND
20637	-121.22	39.43	17	2254	485.0	1007	433	1.7000	INLAND
20638	-121.32	39.43	18	1860	409.0	741	349	1.8672	INLAND
20639	-121.24	39.37	16	2785	616.0	1387	530	2.3886	INLAND

20640 rows × 9 columns

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js