Ml Lab Assignment 2

In [10]:

```
import pandas as pd
import seaborn as sb
import numpy as np
import matplotlib as plt
```

In [11]:

```
data=pd.read_csv(r"C:\Users\stud\Desktop\Dataset\archive\temperatures.csv")
```

In [12]:

```
data.head()
```

Out[12]:

	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	JAN-FEB	MAR-MAY	JUN-SEP	OCT-DEC
0	1901	22.40	24.14	29.07	31.91	33.41	33.18	31.21	30.39	30.47	29.97	27.31	24.49	28.96	23.27	31.46	31.27	27.25
1	1902	24.93	26.58	29.77	31.78	33.73	32.91	30.92	30.73	29.80	29.12	26.31	24.04	29.22	25.75	31.76	31.09	26.49
2	1903	23.44	25.03	27.83	31.39	32.91	33.00	31.34	29.98	29.85	29.04	26.08	23.65	28.47	24.24	30.71	30.92	26.26
3	1904	22.50	24.73	28.21	32.02	32.64	32.07	30.36	30.09	30.04	29.20	26.36	23.63	28.49	23.62	30.95	30.66	26.40
4	1905	22.00	22.83	26.68	30.01	33.32	33.25	31.44	30.68	30.12	30.67	27.52	23.82	28.30	22.25	30.00	31.33	26.57

In [13]:

```
data.isnull().sum()
```

Out[13]:

```
YEAR        0
JAN         0
FEB         0
MAR         0
APR         0
MAY         0
JUN         0
JUL         0
AUG         0
SEP         0
OCT         0
NOV         0
DEC         0
ANNUAL      0
JAN-FEB     0
MAR-MAY     0
JUN-SEP     0
OCT-DEC     0
dtype: int64
```

In [14]:

```
from  sklearn.model_selection import train_test_split
```

In [15]:

```
from sklearn.linear_model import LinearRegression
reg=LinearRegression()
```

In [16]:

```
import matplotlib.pyplot as plt
```

MODEL FOR DECEMBER

In [17]:

```
X=data[["YEAR"]]
Y=data[["DEC"]]
```

In [18]:

```
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.2)
```

In [19]:

```
print(len(x_train))
print(len(x_test))
```

```
93
24
```

In [20]:

```
model=reg.fit(x_train, y_train)
```

In [21]:

```
print(model.coef_)
```

```
[[0.0167988]]
```

In [22]:

```
print(model.intercept_)
```

```
[-8.30440053]
```

In [23]:

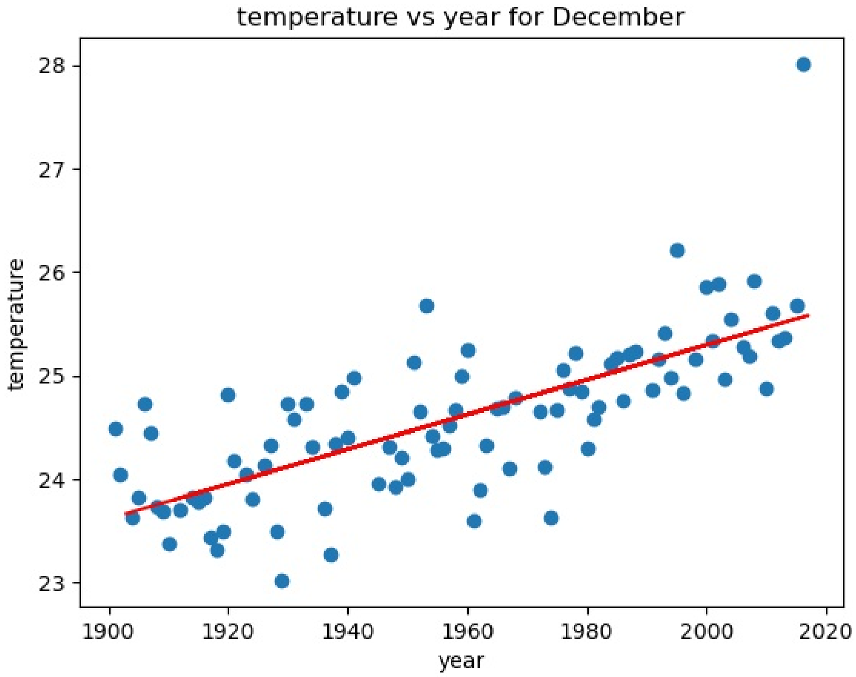
```
y_predict =model.predict(x_test)
```

In [24]:

```
plt.scatter(x_train , y_train)
plt.plot(x_test ,y_predict,color="red")
plt.xlabel("year")
plt.ylabel("temperature")
plt.title("temperature vs year for December")
```

Out[24]:

```
Text(0.5, 1.0, 'temperature vs year for December')
```



In [25]:

```
from sklearn.metrics import mean_squared_error
```

In [26]:

```
from sklearn.metrics import mean_absolute_error
```

In [27]:

```
mean_squared_error(y_test , y_predict)
```

Out[27]:

```
0.29041448828189337
```

In [28]:

```
import math
```

In [29]:

```
mse =mean_squared_error(y_test , y_predict)
```

In [30]:

```
rmse=math.sqrt(mse)
```

In [31]:

```
mae =mean_absolute_error(y_test , y_predict)
```

In [32]:

```
print(f"mae = {mae}")
print(f"mse = {mse}")
print(f"rmse = {rmse}")
```

```
mae = 0.393359515426531\u200b75
mse = 0.29041448828189337
rmse = 0.5389011860089875
```

Model for Feb

In [33]:

```
X=data[["YEAR"]]
Y=data[["FEB"]]
```

In [34]:

```
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.2)
```

In [35]:

```
print(len(x_train))
print(len(x_test))
```

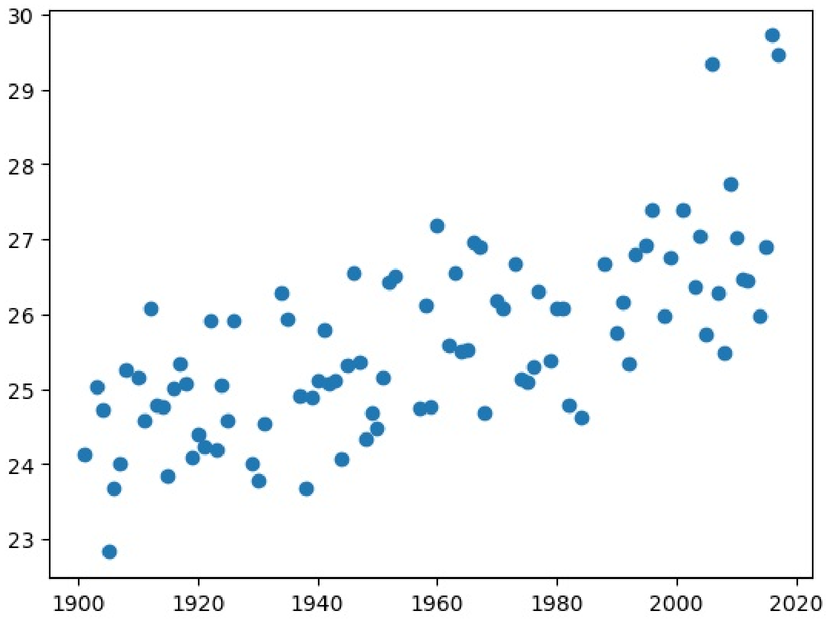
```
93
24
```

In [36]:

```
plt.scatter(x_train , y_train)
```

Out[36]:

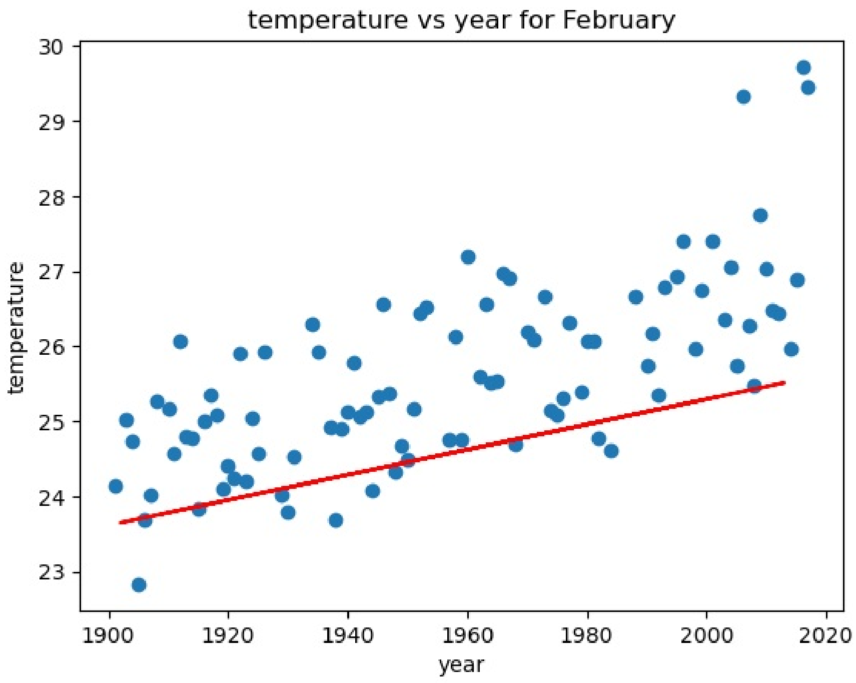
```
<matplotlib.collections.PathCollection at 0x1bd0c36df70>
```



In [37]:
```
y_predict =model.predict(x_test)
```

In [38]:
```
plt.scatter(x_train , y_train)
plt.plot(x_test ,y_predict,color="red")
plt.xlabel("year")
plt.ylabel("temperature")
plt.title("temperature vs year for February")
```

Out[38]:
```
Text(0.5, 1.0, 'temperature vs year for February')
```



In [39]:
```
mean_squared_error(y_test , y_predict)
```

Out[39]:
```
1.1989144489096364
```

In [40]:
```
mae =mean_absolute_error(y_test , y_predict)
mse =mean_squared_error(y_test , y_predict)
```

In [41]:
```
rmse=math.sqrt(mse)
```

In [42]:
```
print(f"mae = {mae}")
print(f"mse = {mse}")
print(f"rmse = {rmse}")
```

```
mae = 0.8993329297029979
mse = 1.1989144489096364
rmse = 1.0949495188864353
```

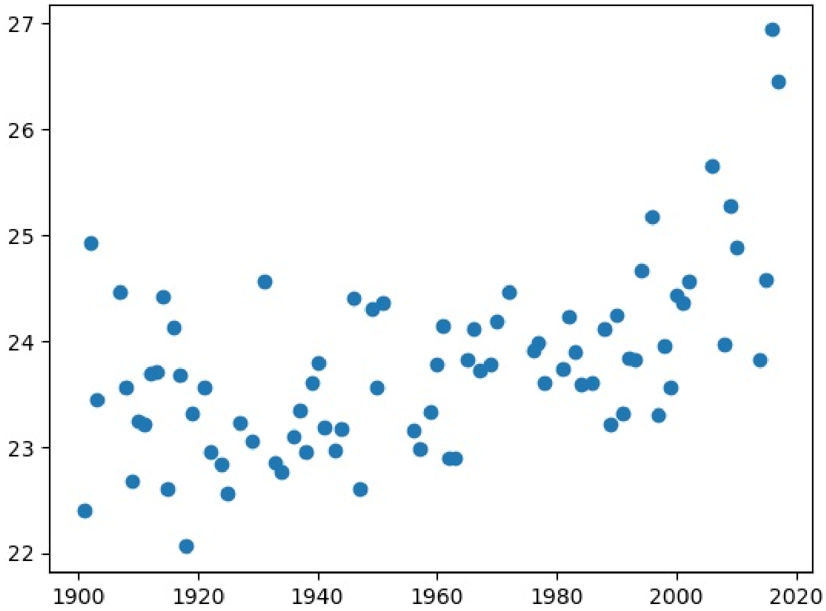
Model For January

In [55]:

```
X=data[["YEAR"]]
Y=data[["JAN"]]
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.3)
plt.scatter(x_train , y_train)
```

Out[55]:

```
<matplotlib.collections.PathCollection at 0x1bd0d9699d0>
```



In [56]:

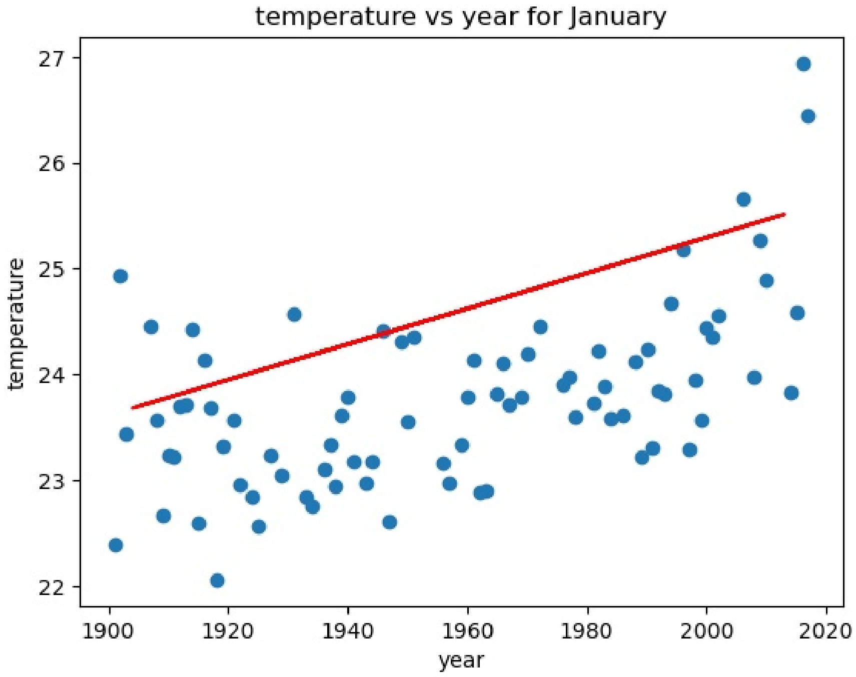
```
y_predict =model.predict(x_test)
```

In [57]:

```
plt.scatter(x_train , y_train)
plt.plot(x_test ,y_predict,color="red")
plt.xlabel("year")
plt.ylabel("temperature")
plt.title("temperature vs year for January")
```

Out[57]:

```
Text(0.5, 1.0, 'temperature vs year for January')
```



In [46]:

```
mean_squared_error(y_test , y_predict)
```

Out[46]:

```
1.513097402242529
```

In [58]:

```
mae =mean_absolute_error(y_test,y_predict)
mse=mean_squared_error(y_test,y_predict)
rsme=math.sqrt(mse)
```

In [59]:

```
print(f"mae= {mae} ")
print(f"mse = {mse}")
print(f"rmse = {rmse}")
```

```
mae= 1.154751518545837
mse = 1.614964556155976
rmse = 1.0949495188864353
```

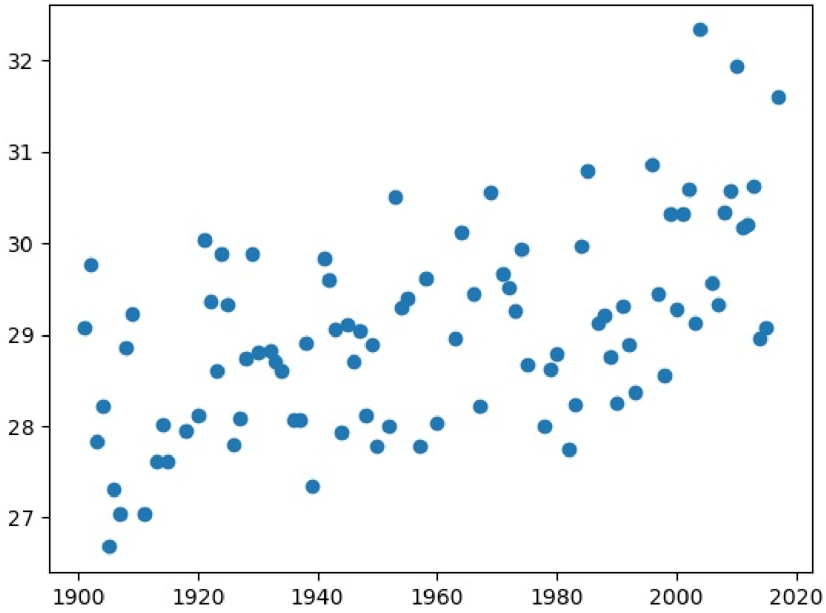
Model for March

In [62]:

```
Y=data[["MAR"]]
X=data[["YEAR"]]
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.2)
plt.scatter(x_train,y_train)
```

Out[62]:

```
<matplotlib.collections.PathCollection at 0x1bd0da19a90>
```



In [63]:

```
model=reg.fit(x_train,y_train)
```

In [64]:

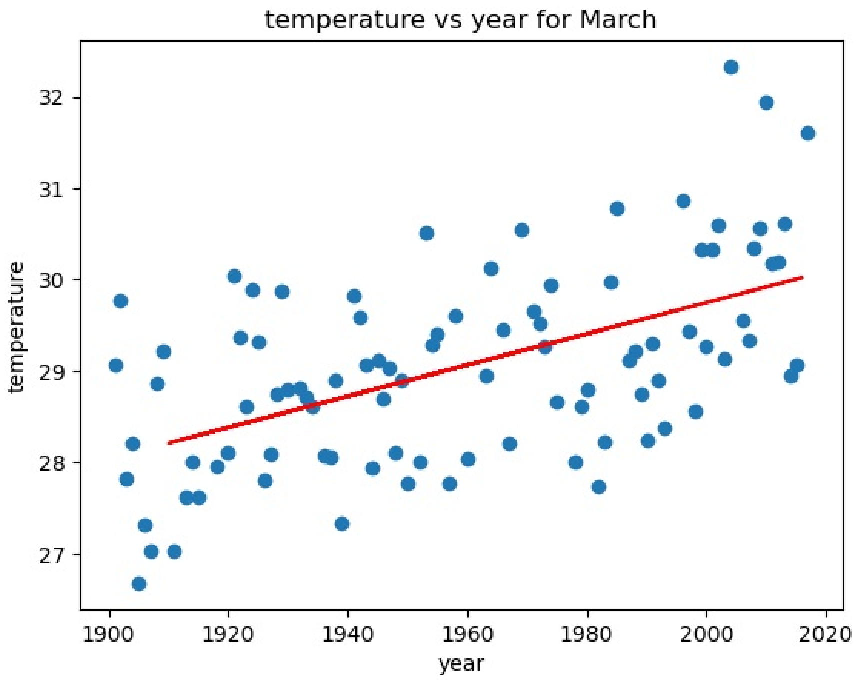
```
y_predict=model.predict(x_test)
```

In [65]:

```
plt.scatter(x_train,y_train)
plt.plot(x_test,y_predict, color="red")
plt.xlabel("year")
plt.ylabel("temperature")
plt.title("temperature vs year for March")
```

Out[65]:

```
Text(0.5, 1.0, 'temperature vs year for March')
```



In [66]:

```
mean_squared_error(y_test , y_predict)
```

Out[66]:

```
0.7080751907494319
```

In [67]:

```
mae =mean_absolute_error(y_test,y_predict)
mse=mean_squared_error(y_test,y_predict)
rsme=math.sqrt(mse)
```

In [68]:

```
print(f"mae= {mae} ")
print(f"mse = {mse}")
print(f"rmse = {rmse}")
```

```
mae= 0.5787098450971695
mse = 0.7080751907494319
rmse = 1.0949495188864353
```

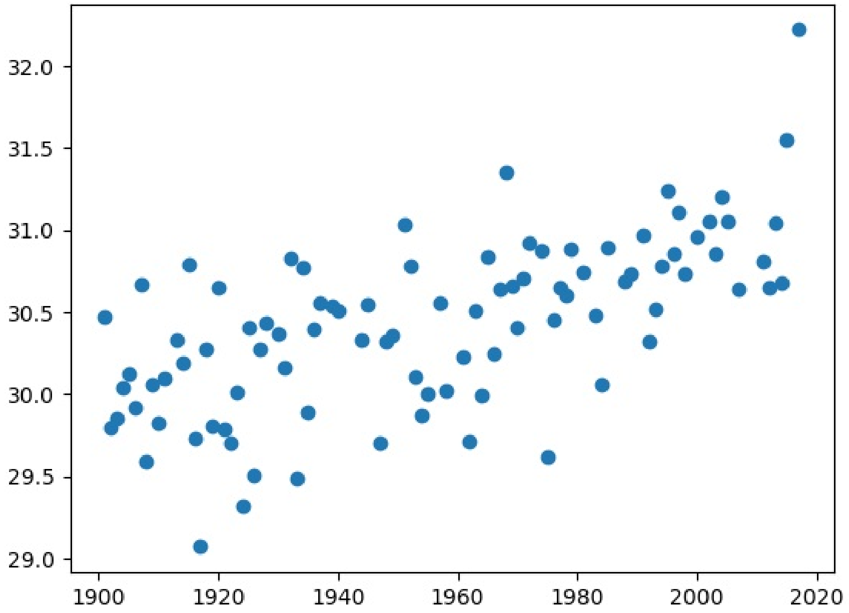
Model for sept

In [70]:

```
Y=data[["SEP"]]
X=data[["YEAR"]]
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.2)
plt.scatter(x_train,y_train)
```

Out[70]:

```
<matplotlib.collections.PathCollection at 0x1bd0dae9c10>
```



In [71]:

```
model=reg.fit(x_train,y_train)
```

In [72]:

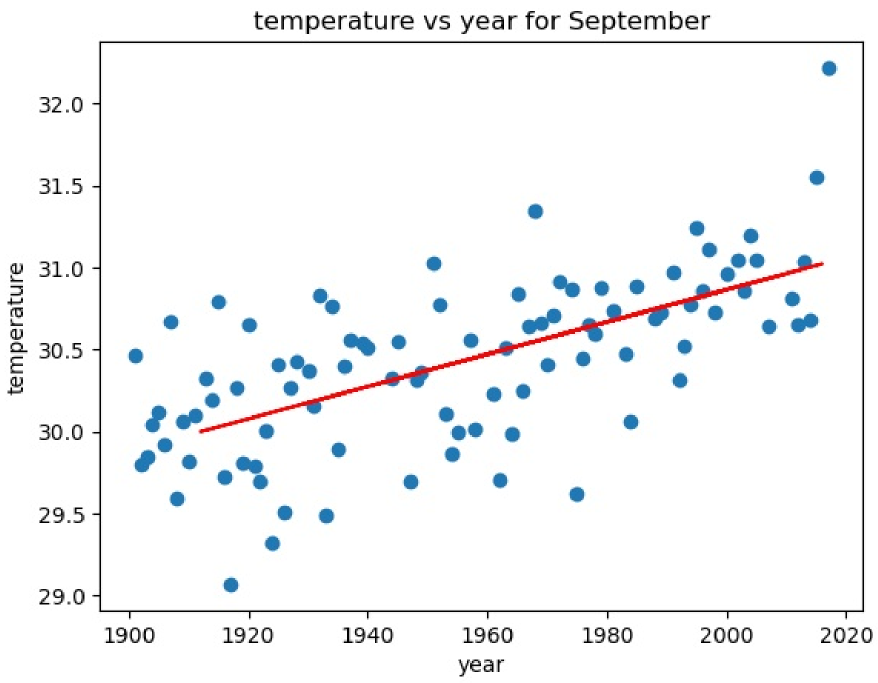
```
y_predict=model.predict(x_test)
```

In [75]:

```
plt.scatter(x_train,y_train)
plt.plot(x_test,y_predict, color="red")
plt.xlabel("year")
plt.ylabel("temperature")
plt.title("temperature vs year for September")
```

Out[75]:

```
Text(0.5, 1.0, 'temperature vs year for September')
```



```
In [76]: mean_squared_error(y_test , y_predict)

Out[76]: 0.20095521958128926
```

```
In [77]: mae =mean_absolute_error(y_test,y_predict)
         mse=mean_squared_error(y_test,y_predict)
         rsme=math.sqrt(mse)
```

```
In [78]: print(f"mae= {mae} ")
         print(f"mse = {mse}")
         print(f"rmse = {rmse}")

         mae= 0.3369482708223573
         mse = 0.20095521958128926
         rmse = 1.0949495188864353
```

```
In [ ]:
```

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