In [52]:
```python
import pandas as pd
import numpy as np
```

In [53]:
```python
df = pd.read_csv('webtraffic.csv')
```

In [54]:
```python
df.head()
```

Out[54]:

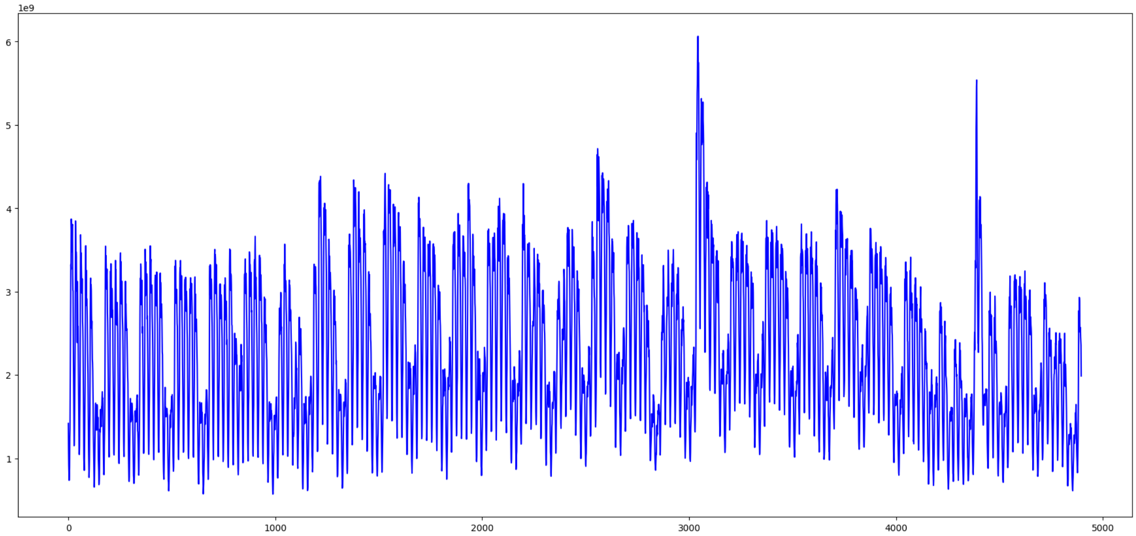
	Hour Index	Sessions
0	0	1418159421
1	1	1113769116
2	2	919158921
3	3	822352824
4	4	735526737

In [55]:
```python
df.shape
```

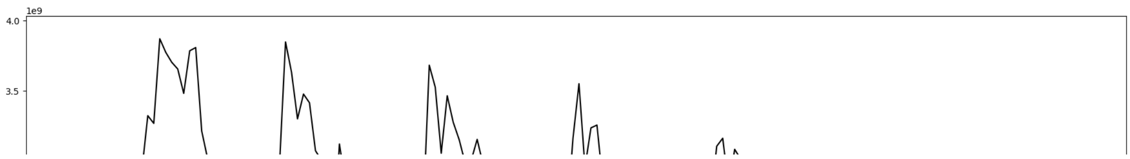
Out[55]: (4896, 2)

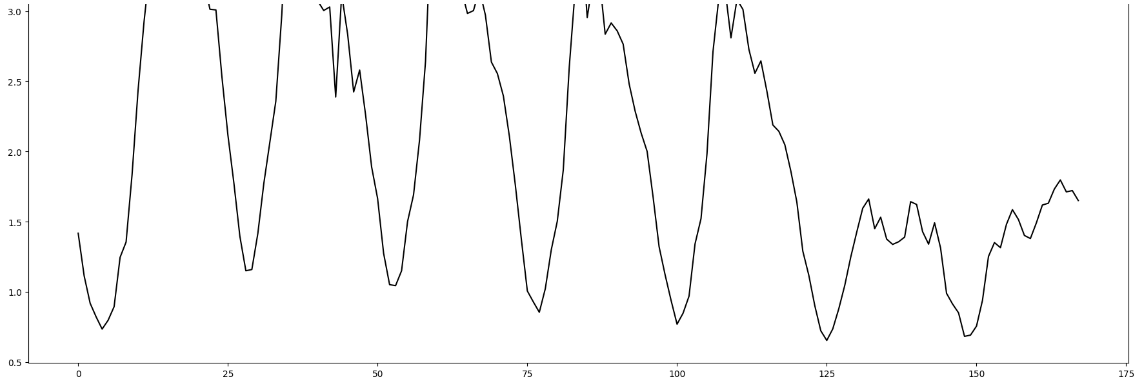
Data Exploration for Web Traffic Forecasting

In [56]:
```python
import matplotlib.pyplot as plt
sessions = df['Sessions'].values
ar = np.arange(len(sessions))
plt.figure(figsize=(22,10))
plt.plot(ar, sessions,'b')
plt.show()
```



In [57]:
```python
#first week web traffic
sample = sessions[:168]
ar = np.arange(len(sample))
plt.figure(figsize=(22,10))
plt.plot(ar, sample,'black')
plt.show()
```





Data Preparation for Web Traffic Forecasting

In [58]:
```
def prepare_data(seq,num):
  x=[]
  y=[]
  for i in range(0,(len(seq)-num),1):

    input_ = seq[i:i+num]
    output  = seq[i+num]

    x.append(input_)
    y.append(output)

  return np.array(x), np.array(y)
```

In [59]:
```
num=168 #1 week = 168hrs
x,y= prepare_data(sessions,num)
print(len(x))
```

4728

Split the Dataset

In [60]:
```
ind = int(0.9 * len(x)) #split into 90:10
x_tr = x[:ind]
y_tr = y[:ind]
x_val=x[ind:]
y_val=y[ind:]
```

In [61]:
```
from sklearn.preprocessing import StandardScaler
#normalize the inputs
x_scaler= StandardScaler()
x_tr = x_scaler.fit_transform(x_tr)
x_val= x_scaler.transform(x_val)
#reshaping the output for normalization
y_tr=y_tr.reshape(len(y_tr),1)
y_val=y_val.reshape(len(y_val),1)
#normalize the output
y_scaler=StandardScaler()
y_tr = y_scaler.fit_transform(y_tr)[:,0]
y_val = y_scaler.transform(y_val)[:,0]
```

In [62]:
```
#reshaping input data
x_tr= x_tr.reshape(x_tr.shape[0],x_tr.shape[1],1)
x_val= x_val.reshape(x_val.shape[0],x_val.shape[1],1)
print(x_tr.shape)
```

(4255, 168, 1)

Model Building for Web Traffic Forecasting

In [63]:
```
from keras.models import *
```

```python
from keras.layers import *
from keras.callbacks import *
from tensorflow import keras
# define model
model =  Sequential()
model.add(LSTM(128,input_shape=(168,1)))
model.add(Dense(64,activation='relu'))
model.add(Dense(1,activation='linear'))
```

In [64]:

```python
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 lstm_1 (LSTM)               (None, 128)               66560     
                                                                 
 dense_4 (Dense)             (None, 64)                8256      
                                                                 
 dense_5 (Dense)             (None, 1)                 65        
                                                                 
=================================================================
Total params: 74,881
Trainable params: 74,881
Non-trainable params: 0
_________________________________________________________________
```

In [65]:

```python
# Define the optimizer and loss
model.compile(loss='mse',optimizer='adam')
#Define the callback to save the best model during the training
mc = ModelCheckpoint('best_model.hdf5', monitor='val_loss',
        verbose=1, save_best_only=True, mode='min')
# Train the model for 30 epochs with batch size of 32:
history=model.fit(x_tr, y_tr ,epochs=30, batch_size=32,
           validation_data=(x_val,y_val), callbacks=[mc])
```

```
Epoch 1/30
133/133 [==============================] - ETA: 0s - loss: 0.1544
Epoch 1: val_loss improved from inf to 0.03151, saving model to best_model.hdf5
133/133 [==============================] - 32s 191ms/step - loss: 0.1544 - val_loss: 0.031
5
Epoch 2/30
133/133 [==============================] - ETA: 0s - loss: 0.0399
Epoch 2: val_loss did not improve from 0.03151
133/133 [==============================] - 19s 141ms/step - loss: 0.0399 - val_loss: 0.036
2
Epoch 3/30
133/133 [==============================] - ETA: 0s - loss: 0.0350
Epoch 3: val_loss improved from 0.03151 to 0.03004, saving model to best_model.hdf5
133/133 [==============================] - 18s 136ms/step - loss: 0.0350 - val_loss: 0.030
0
Epoch 4/30
133/133 [==============================] - ETA: 0s - loss: 0.0332
Epoch 4: val_loss improved from 0.03004 to 0.02779, saving model to best_model.hdf5
133/133 [==============================] - 24s 182ms/step - loss: 0.0332 - val_loss: 0.027
8
Epoch 5/30
133/133 [==============================] - ETA: 0s - loss: 0.0328
Epoch 5: val_loss improved from 0.02779 to 0.02505, saving model to best_model.hdf5
133/133 [==============================] - 31s 233ms/step - loss: 0.0328 - val_loss: 0.025
0
Epoch 6/30
133/133 [==============================] - ETA: 0s - loss: 0.0302
Epoch 6: val_loss did not improve from 0.02505
133/133 [==============================] - 25s 189ms/step - loss: 0.0302 - val_loss: 0.027
2
Epoch 7/30
133/133 [==============================] - ETA: 0s - loss: 0.0274
Epoch 7: val_loss improved from 0.02505 to 0.02405, saving model to best_model.hdf5
133/133 [==============================] - 24s 183ms/step - loss: 0.0274 - val_loss: 0.024
1
Epoch 8/30
```

```
Epoch 8/30
133/133 [==============================] - ETA: 0s - loss: 0.0276
Epoch 8: val_loss improved from 0.02405 to 0.02384, saving model to best_model.hdf5
133/133 [==============================] - 25s 189ms/step - loss: 0.0276 - val_loss: 0.023
8
Epoch 9/30
133/133 [==============================] - ETA: 0s - loss: 0.0261
Epoch 9: val_loss did not improve from 0.02384
133/133 [==============================] - 24s 179ms/step - loss: 0.0261 - val_loss: 0.025
5
Epoch 10/30
133/133 [==============================] - ETA: 0s - loss: 0.0248
Epoch 10: val_loss did not improve from 0.02384
133/133 [==============================] - 31s 233ms/step - loss: 0.0248 - val_loss: 0.024
5
Epoch 11/30
133/133 [==============================] - ETA: 0s - loss: 0.0217
Epoch 11: val_loss did not improve from 0.02384
133/133 [==============================] - 27s 203ms/step - loss: 0.0217 - val_loss: 0.025
1
Epoch 12/30
133/133 [==============================] - ETA: 0s - loss: 0.0200
Epoch 12: val_loss improved from 0.02384 to 0.01826, saving model to best_model.hdf5
133/133 [==============================] - 23s 176ms/step - loss: 0.0200 - val_loss: 0.018
3
Epoch 13/30
133/133 [==============================] - ETA: 0s - loss: 0.0201
Epoch 13: val_loss improved from 0.01826 to 0.01791, saving model to best_model.hdf5
133/133 [==============================] - 23s 174ms/step - loss: 0.0201 - val_loss: 0.017
9
Epoch 14/30
133/133 [==============================] - ETA: 0s - loss: 0.0191
Epoch 14: val_loss did not improve from 0.01791
133/133 [==============================] - 21s 160ms/step - loss: 0.0191 - val_loss: 0.019
9
Epoch 15/30
133/133 [==============================] - ETA: 0s - loss: 0.0186
Epoch 15: val_loss improved from 0.01791 to 0.01669, saving model to best_model.hdf5
133/133 [==============================] - 21s 155ms/step - loss: 0.0186 - val_loss: 0.016
7
Epoch 16/30
133/133 [==============================] - ETA: 0s - loss: 0.0174
Epoch 16: val_loss improved from 0.01669 to 0.01448, saving model to best_model.hdf5
133/133 [==============================] - 20s 151ms/step - loss: 0.0174 - val_loss: 0.014
5
Epoch 17/30
133/133 [==============================] - ETA: 0s - loss: 0.0182
Epoch 17: val_loss did not improve from 0.01448
133/133 [==============================] - 19s 146ms/step - loss: 0.0182 - val_loss: 0.017
2
Epoch 18/30
133/133 [==============================] - ETA: 0s - loss: 0.0176
Epoch 18: val_loss did not improve from 0.01448
133/133 [==============================] - 20s 152ms/step - loss: 0.0176 - val_loss: 0.019
0
Epoch 19/30
133/133 [==============================] - ETA: 0s - loss: 0.0163
Epoch 19: val_loss did not improve from 0.01448
133/133 [==============================] - 20s 149ms/step - loss: 0.0163 - val_loss: 0.016
0
Epoch 20/30
133/133 [==============================] - ETA: 0s - loss: 0.0167
Epoch 20: val_loss did not improve from 0.01448
133/133 [==============================] - 26s 199ms/step - loss: 0.0167 - val_loss: 0.016
6
Epoch 21/30
133/133 [==============================] - ETA: 0s - loss: 0.0161
Epoch 21: val_loss did not improve from 0.01448
133/133 [==============================] - 27s 204ms/step - loss: 0.0161 - val_loss: 0.016
8
Epoch 22/30
133/133 [==============================] - ETA: 0s - loss: 0.0164
Epoch 22: val_loss did not improve from 0.01448
133/133 [==============================] - 19s 145ms/step - loss: 0.0164 - val_loss: 0.016
5
```

```
Epoch 23/30
133/133 [==============================] - ETA: 0s - loss: 0.0164
Epoch 23: val_loss did not improve from 0.01448
133/133 [==============================] - 19s 144ms/step - loss: 0.0164 - val_loss: 0.019
2
Epoch 24/30
133/133 [==============================] - ETA: 0s - loss: 0.0162
Epoch 24: val_loss did not improve from 0.01448
133/133 [==============================] - 19s 145ms/step - loss: 0.0162 - val_loss: 0.023
0
Epoch 25/30
133/133 [==============================] - ETA: 0s - loss: 0.0180
Epoch 25: val_loss did not improve from 0.01448
133/133 [==============================] - 19s 143ms/step - loss: 0.0180 - val_loss: 0.016
6
Epoch 26/30
133/133 [==============================] - ETA: 0s - loss: 0.0155
Epoch 26: val_loss did not improve from 0.01448
133/133 [==============================] - 19s 144ms/step - loss: 0.0155 - val_loss: 0.015
9
Epoch 27/30
133/133 [==============================] - ETA: 0s - loss: 0.0158
Epoch 27: val_loss did not improve from 0.01448
133/133 [==============================] - 19s 146ms/step - loss: 0.0158 - val_loss: 0.018
8
Epoch 28/30
133/133 [==============================] - ETA: 0s - loss: 0.0153
Epoch 28: val_loss did not improve from 0.01448
133/133 [==============================] - 19s 144ms/step - loss: 0.0153 - val_loss: 0.015
4
Epoch 29/30
133/133 [==============================] - ETA: 0s - loss: 0.0152
Epoch 29: val_loss improved from 0.01448 to 0.01438, saving model to best_model.hdf5
133/133 [==============================] - 19s 145ms/step - loss: 0.0152 - val_loss: 0.014
4
Epoch 30/30
133/133 [==============================] - ETA: 0s - loss: 0.0160
Epoch 30: val_loss did not improve from 0.01438
133/133 [==============================] - 18s 138ms/step - loss: 0.0160 - val_loss: 0.020
1
```

In [66]:

```
model.load_weights('best_model.hdf5')
```

In [67]:

```
mse = model.evaluate(x_val,y_val)
print("Mean Square Error:",mse)
```

```
 4/15 [=======>......................] - ETA: 0s - loss: 0.018515/15 [====================
==========] - 1s 64ms/step - loss: 0.0144
Mean Square Error: 0.014384998008608818
```

Baseline Model with Forecasting

In [68]:

```
# build a simple moving average model
def compute_moving_average(data):
  pred=[]
  for i in data:
    avg=np.sum(i)/len(i)
    pred.append(avg)
  return np.array(pred)
# reshape the data
x_reshaped = x_val.reshape(-1,168)
# get predictions
y_pred = compute_moving_average(x_reshaped)
# evaluate the performance of model on the validation data
mse = np.sum ( (y_val - y_pred) **2 ) / (len(y_val))
print("Mean square of error:- ",mse)
```

```
Mean square of error:-  0.5546025834434455
```

Web Traffic Forecasting

In [69]:

```python
def forecast(x_val, no_of_pred, ind):
  predictions=[]
  #intialize the array with a weeks data
  temp=x_val[ind]
  for i in range(no_of_pred):
    #predict for the next hour
    pred=model.predict(temp.reshape(1,-1,1))[0][0]

    #append the prediction as the last element of array
    temp = np.insert(temp,len(temp),pred)
    predictions.append(pred)
    #ignore the first element of array
    temp = temp[1:]
  return predictions
```

In [70]:

```python
no_of_pred =24
ind=72
y_pred= forecast(x_val,no_of_pred,ind)
y_true = y_val[ind:ind+(no_of_pred)]
```

```
1/1 [==============================] - 1s 920ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 40ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 26ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 32ms/step
```

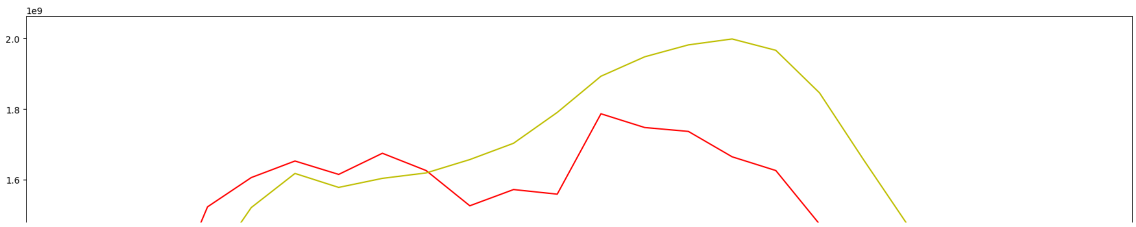
In [71]:

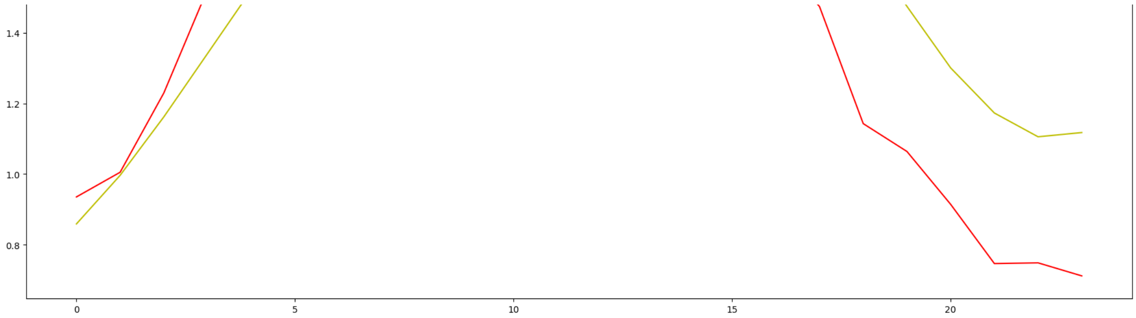
```python
y_true = np.array(y_true)
y_true = y_true.reshape(-1, 1)

y_pred = np.array(y_pred)
y_pred = y_pred.reshape(-1, 1)
y_true= y_scaler.inverse_transform(y_true)
y_pred= y_scaler.inverse_transform(y_pred)
```

In [72]:

```python
def plot(y_true,y_pred):
  ar = np.arange(len(y_true))
  plt.figure(figsize=(22,10))
  plt.plot(ar, y_true,'r')
  plt.plot(ar, y_pred,'y')
  plt.show()
plot(y_true,y_pred)
```





CNN Model with Forecasting

In [73]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import *
from tensorflow.keras.callbacks import *
model= Sequential()
model.add(Conv1D(64, 3, padding='same', activation='relu',input_shape=(num,1)))
model.add(Conv1D(32, 5, padding='same', activation='relu',input_shape=(num,1)))
model.add(Flatten())
model.add(Dense(64,activation='relu'))
model.add(Dense(1,activation='linear'))
model.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv1d_2 (Conv1D)           (None, 168, 64)           256       
                                                                 
 conv1d_3 (Conv1D)           (None, 168, 32)           10272     
                                                                 
 flatten_1 (Flatten)         (None, 5376)              0         
                                                                 
 dense_6 (Dense)             (None, 64)                344128    
                                                                 
 dense_7 (Dense)             (None, 1)                 65        
                                                                 
=================================================================
Total params: 354,721
Trainable params: 354,721
Non-trainable params: 0
_________________________________________________________________
```

In [74]:
```
# Define the optimizer and loss:
model.compile(loss='mse',optimizer='adam')
# Define the callback to save the best model during the training
mc = ModelCheckpoint('best_model.hdf5', monitor='val_loss', verbose=1,
        save_best_only=True, mode='min')
# Train the model for 30 epochs with batch size of 32:
history=model.fit(x_tr, y_tr ,epochs=30, batch_size=32, validation_data=(x_val,y_val),
            callbacks=[mc])
```

```
Epoch 1/30
133/133 [==============================] - ETA: 0s - loss: 0.0971
Epoch 1: val_loss improved from inf to 0.04090, saving model to best_model.hdf5
133/133 [==============================] - 3s 19ms/step - loss: 0.0971 - val_loss: 0.0409
Epoch 2/30
131/133 [============================>.] - ETA: 0s - loss: 0.0238
Epoch 2: val_loss improved from 0.04090 to 0.02447, saving model to best_model.hdf5
133/133 [==============================] - 3s 21ms/step - loss: 0.0238 - val_loss: 0.0245
Epoch 3/30
131/133 [============================>.] - ETA: 0s - loss: 0.0172
Epoch 3: val_loss improved from 0.02447 to 0.01884, saving model to best_model.hdf5
133/133 [==============================] - 3s 20ms/step - loss: 0.0174 - val_loss: 0.0188
Epoch 4/30
130/133 [============================>.] - ETA: 0s - loss: 0.0167
Epoch 4: val_loss improved from 0.01884 to 0.01482, saving model to best_model.hdf5
133/133 [==============================] - 3s 20ms/step - loss: 0.0165 - val_loss: 0.0148
Epoch 5/30
131/133 [============================>.] - ETA: 0s - loss: 0.0127
```

```
Epoch 5: val_loss did not improve from 0.01482
133/133 [==============================] - 3s 19ms/step - loss: 0.0126 - val_loss: 0.0164
Epoch 6/30
132/133 [============================>.] - ETA: 0s - loss: 0.0118
Epoch 6: val_loss improved from 0.01482 to 0.01394, saving model to best_model.hdf5
133/133 [==============================] - 3s 20ms/step - loss: 0.0119 - val_loss: 0.0139
Epoch 7/30
131/133 [============================>.] - ETA: 0s - loss: 0.0112
Epoch 7: val_loss did not improve from 0.01394
133/133 [==============================] - 3s 20ms/step - loss: 0.0112 - val_loss: 0.0155
Epoch 8/30
133/133 [==============================] - ETA: 0s - loss: 0.0105
Epoch 8: val_loss improved from 0.01394 to 0.01391, saving model to best_model.hdf5
133/133 [==============================] - 3s 20ms/step - loss: 0.0105 - val_loss: 0.0139
Epoch 9/30
133/133 [==============================] - ETA: 0s - loss: 0.0105
Epoch 9: val_loss did not improve from 0.01391
133/133 [==============================] - 3s 21ms/step - loss: 0.0105 - val_loss: 0.0140
Epoch 10/30
132/133 [============================>.] - ETA: 0s - loss: 0.0087
Epoch 10: val_loss improved from 0.01391 to 0.01337, saving model to best_model.hdf5
133/133 [==============================] - 3s 21ms/step - loss: 0.0087 - val_loss: 0.0134
Epoch 11/30
133/133 [==============================] - ETA: 0s - loss: 0.0084
Epoch 11: val_loss did not improve from 0.01337
133/133 [==============================] - 3s 19ms/step - loss: 0.0084 - val_loss: 0.0147
Epoch 12/30
131/133 [============================>.] - ETA: 0s - loss: 0.0090
Epoch 12: val_loss did not improve from 0.01337
133/133 [==============================] - 3s 19ms/step - loss: 0.0090 - val_loss: 0.0137
Epoch 13/30
132/133 [============================>.] - ETA: 0s - loss: 0.0078
Epoch 13: val_loss did not improve from 0.01337
133/133 [==============================] - 3s 20ms/step - loss: 0.0078 - val_loss: 0.0137
Epoch 14/30
131/133 [============================>.] - ETA: 0s - loss: 0.0064
Epoch 14: val_loss did not improve from 0.01337
133/133 [==============================] - 3s 19ms/step - loss: 0.0064 - val_loss: 0.0146
Epoch 15/30
132/133 [============================>.] - ETA: 0s - loss: 0.0059
Epoch 15: val_loss did not improve from 0.01337
133/133 [==============================] - 2s 19ms/step - loss: 0.0059 - val_loss: 0.0153
Epoch 16/30
131/133 [============================>.] - ETA: 0s - loss: 0.0059
Epoch 16: val_loss did not improve from 0.01337
133/133 [==============================] - 3s 19ms/step - loss: 0.0058 - val_loss: 0.0167
Epoch 17/30
131/133 [============================>.] - ETA: 0s - loss: 0.0049
Epoch 17: val_loss did not improve from 0.01337
133/133 [==============================] - 3s 19ms/step - loss: 0.0049 - val_loss: 0.0152
Epoch 18/30
131/133 [============================>.] - ETA: 0s - loss: 0.0043
Epoch 18: val_loss did not improve from 0.01337
133/133 [==============================] - 2s 19ms/step - loss: 0.0043 - val_loss: 0.0165
Epoch 19/30
133/133 [==============================] - ETA: 0s - loss: 0.0041
Epoch 19: val_loss did not improve from 0.01337
133/133 [==============================] - 3s 20ms/step - loss: 0.0041 - val_loss: 0.0164
Epoch 20/30
133/133 [==============================] - ETA: 0s - loss: 0.0038
Epoch 20: val_loss did not improve from 0.01337
133/133 [==============================] - 3s 19ms/step - loss: 0.0038 - val_loss: 0.0155
Epoch 21/30
132/133 [============================>.] - ETA: 0s - loss: 0.0033
Epoch 21: val_loss did not improve from 0.01337
133/133 [==============================] - 3s 19ms/step - loss: 0.0033 - val_loss: 0.0163
Epoch 22/30
133/133 [==============================] - ETA: 0s - loss: 0.0028
Epoch 22: val_loss did not improve from 0.01337
133/133 [==============================] - 2s 19ms/step - loss: 0.0028 - val_loss: 0.0164
Epoch 23/30
132/133 [============================>.] - ETA: 0s - loss: 0.0025
Epoch 23: val_loss did not improve from 0.01337
133/133 [==============================] - 3s 19ms/step - loss: 0.0025 - val_loss: 0.0165
```

```
Epoch 24/30
133/133 [==============================] - ETA: 0s - loss: 0.0035
Epoch 24: val_loss did not improve from 0.01337
133/133 [==============================] - 3s 19ms/step - loss: 0.0035 - val_loss: 0.0155
Epoch 25/30
131/133 [============================>.] - ETA: 0s - loss: 0.0030
Epoch 25: val_loss did not improve from 0.01337
133/133 [==============================] - 2s 19ms/step - loss: 0.0030 - val_loss: 0.0158
Epoch 26/30
131/133 [============================>.] - ETA: 0s - loss: 0.0019
Epoch 26: val_loss did not improve from 0.01337
133/133 [==============================] - 3s 19ms/step - loss: 0.0019 - val_loss: 0.0174
Epoch 27/30
132/133 [============================>.] - ETA: 0s - loss: 0.0018
Epoch 27: val_loss did not improve from 0.01337
133/133 [==============================] - 3s 20ms/step - loss: 0.0018 - val_loss: 0.0156
Epoch 28/30
131/133 [============================>.] - ETA: 0s - loss: 0.0016
Epoch 28: val_loss did not improve from 0.01337
133/133 [==============================] - 3s 19ms/step - loss: 0.0016 - val_loss: 0.0179
Epoch 29/30
132/133 [============================>.] - ETA: 0s - loss: 0.0013
Epoch 29: val_loss did not improve from 0.01337
133/133 [==============================] - 3s 19ms/step - loss: 0.0013 - val_loss: 0.0176
Epoch 30/30
131/133 [============================>.] - ETA: 0s - loss: 0.0015
Epoch 30: val_loss did not improve from 0.01337
133/133 [==============================] - 3s 19ms/step - loss: 0.0015 - val_loss: 0.0154
```

In [75]:

```python
model.load_weights('best_model.hdf5')
```

In [76]:

```python
mse = model.evaluate(x_val,y_val)
print("Mean Square Error:",mse)
```

```
15/15 [==============================] - 0s 5ms/step - loss: 0.0134
Mean Square Error: 0.013367628678679466
```

In [77]:

```python
#build a simple model
def compute_moving_average(data):
  pred=[]
  for i in data:
    avg=np.sum(i)/len(i)
    pred.append(avg)
  return np.array(pred)
x_reshaped = x_val.reshape(-1,168)
y_pred = compute_moving_average(x_reshaped)
mse = np.sum ( (y_val - y_pred) **2 ) / (len(y_val))
print("Mean Square Error:",mse)
```

```
Mean Square Error: 0.5546025834434455
```

In [78]:

```python
def forecast(x_val, no_of_pred, ind):
  predictions=[]
  #intialize the array with previous weeks data
  temp=x_val[ind]
  for i in range(no_of_pred):
    #predict for the next hour
    pred=model.predict(temp.reshape(1,-1,1))[0][0]

    #append the prediction as the last element of array
    temp = np.insert(temp,len(temp),pred)
    predictions.append(pred)
    #ignore the first element of array
    temp = temp[1:]
  return predictions
```

In [79]:

```python
no_of_pred =24
ind=72
y_pred= forecast(x_val,no_of_pred,ind)
```

```
y_true = y_val[ind:ind+(no_of_pred)]
```

```
1/1 [==============================] - 0s 181ms/step
1/1 [==============================] - 0s 43ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 46ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 40ms/step
1/1 [==============================] - 0s 42ms/step
1/1 [==============================] - 0s 43ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 43ms/step
1/1 [==============================] - 0s 37ms/step
1/1 [==============================] - 0s 43ms/step
1/1 [==============================] - 0s 44ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 42ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 42ms/step
1/1 [==============================] - 0s 46ms/step
1/1 [==============================] - 0s 42ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 40ms/step
```

In [80]:

```
y_true = np.array(y_true)
y_true = y_true.reshape(-1, 1)

y_pred = np.array(y_pred)
y_pred = y_pred.reshape(-1, 1)
y_true= y_scaler.inverse_transform(y_true)
y_pred= y_scaler.inverse_transform(y_pred)
```

In [81]:

```
def plot(y_true,y_pred):
  ar = np.arange(len(y_true))
  plt.figure(figsize=(22,10))
  plt.plot(ar, y_true,'r')
  plt.plot(ar, y_pred,'y')
  plt.show()
```

In [82]:

```
plot(y_true,y_pred)
```