

# Powerball Dataset Guide

This document contains the steps to separate the space separated Winning Numbers into separate Integers, and a short guide to time series models and time series analysis.

## How to separate the space separated Winning Numbers into separate Integers

- Let us first start with the dataset. Once you have imported/downloaded the dataset from Kaggle and renamed it , run a `pd.read_csv('powerball.csv')` , and then a `data.head()` . We see that the Winning Numbers column is a space separated string of numbers. Let us view the column to see it's type. We find that it is of object type.

```
[47] data = pd.read_csv('powerball.csv')

data.head()
```

	Draw Date	Winning Numbers	Multiplier
0	09/26/2020	11 21 27 36 62 24	3.0
1	09/30/2020	14 18 36 49 67 18	2.0
2	10/03/2020	18 31 36 43 47 20	2.0
3	10/07/2020	06 24 30 53 56 19	2.0
4	10/10/2020	05 18 23 40 50 18	3.0

```
[49] data['Winning Numbers']
```


```
0      11 21 27 36 62 24
1      14 18 36 49 67 18
2      18 31 36 43 47 20
3       06 24 30 53 56 19
4       05 18 23 40 50 18
...
1424    01 04 12 36 49 05
1425    31 43 58 59 66 09
1426    02 08 15 19 58 10
1427    05 11 22 23 69 07
1428    52 58 59 64 66 09
Name: Winning Numbers, Length: 1429, dtype: object
```

- Now, to convert this space separated string to separate columns of type integer, we can run the following code:

```
[50] data['Winning Numbers'] = data['Winning Numbers'].str.split(' ')
data[['ball1', 'ball2', 'ball3', 'ball4', 'ball5', 'ball6']] = data['Winning Numbers'].apply(lambda x: pd.Series(x))
data[['ball1', 'ball2', 'ball3', 'ball4', 'ball5', 'ball6']] = data[['ball1', 'ball2', 'ball3', 'ball4', 'ball5', 'ball6']].astype(int)
```

- `data['Winning Numbers'] = data['Winning Numbers'].str.split(' ')` - This line splits the values in the "Winning Numbers" column by the space character (' ') using the split() method from the pandas Series str accessor. The result is a new Series where each value is a list of strings representing the individual numbers.`
- `data[['ball1', 'ball2', 'ball3', 'ball4', 'ball5', 'ball6']] = data['Winning Numbers'].apply(lambda x: pd.Series(x))` - This line uses the apply() method to apply a lambda function to each value in the "Winning Numbers" column. The lambda function converts each list of numbers into a pandas Series. The result is a DataFrame with six new columns ('ball1', 'ball2', 'ball3', 'ball4', 'ball5', 'ball6'), each containing the corresponding number from the original list.`
- `data[['ball1', 'ball2', 'ball3', 'ball4', 'ball5', 'ball6']] = data[['ball1', 'ball2', 'ball3', 'ball4', 'ball5', 'ball6']].astype(int)` - This line converts the data type of the six new columns ('ball1', 'ball2', 'ball3', 'ball4', 'ball5', 'ball6') to integer using the astype() method. This ensures that the extracted numbers are represented as numerical values rather than strings.`
- In summary, these three lines of code split the values in the "Winning Numbers" column, create separate columns for each number, and convert the data type of these columns to integer for further analysis and processing.

- To see what the data looks like now, we can run a `data.head()` again.:

 `data.head()`

	Draw Date	Winning Numbers	Multiplier	ball1	ball2	ball3	ball4	ball5	ball6
0	09/26/2020	[11, 21, 27, 36, 62, 24]	3.0	11	21	27	36	62	24
1	09/30/2020	[14, 18, 36, 49, 67, 18]	2.0	14	18	36	49	67	18
2	10/03/2020	[18, 31, 36, 43, 47, 20]	2.0	18	31	36	43	47	20
3	10/07/2020	[06, 24, 30, 53, 56, 19]	2.0	6	24	30	53	56	19
4	10/10/2020	[05, 18, 23, 40, 50, 18]	3.0	5	18	23	40	50	18

- And to check if the data got converted to Integer correctly, we can run a `data.info()`:

```
✓ [53] data.info()
0s
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1429 entries, 0 to 1428
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Draw Date             1429 non-null   object
1   Winning Numbers       1429 non-null   object
2   Multiplier            1219 non-null   float64
3   ball1                 1429 non-null   int64
4   ball2                 1429 non-null   int64
5   ball3                 1429 non-null   int64
6   ball4                 1429 non-null   int64
7   ball5                 1429 non-null   int64
8   ball6                 1429 non-null   int64
dtypes: float64(1), int64(6), object(2)
memory usage: 100.6+ KB
```

- Now you can start analysing this data and creating a model!

# How to do Time Series Analysis, and pick your model

Time series models are designed to analyze and make predictions based on patterns and trends in sequential data over time. Here are a few examples of time series models that can be used with the Powerball data:

- 1. ARIMA (AutoRegressive Integrated Moving Average):** ARIMA models are widely used for time series analysis and forecasting. They incorporate autoregressive (AR), moving average (MA), and differencing (I) components. ARIMA models can capture patterns and seasonality in the data. You can use historical Powerball data to train an ARIMA model and make predictions for future winning numbers.
- 2. Seasonal ARIMA (SARIMA):** SARIMA models extend the ARIMA model to incorporate seasonality in the data. They are suitable for time series with recurring patterns at fixed intervals. If you observe seasonality in the Powerball data, you can consider applying a SARIMA model to capture and forecast these seasonal patterns.
- 3. Prophet:** Prophet is a time series forecasting library developed by Facebook. It is designed to handle time series data with various trends, seasonality, and holidays. Prophet models are relatively easy to implement and provide robust forecasts. You can use Prophet to analyze and predict future winning numbers based on historical Powerball data.
- 4. Long Short-Term Memory (LSTM) Networks:** LSTM is a type of recurrent neural network (RNN) that can model sequences and capture long-term dependencies. LSTM networks are well-suited for analyzing and predicting time series data. You can train an LSTM network using the historical Powerball data and use it to forecast future winning numbers.

These are just a few examples of time series models that you can apply to the Powerball data. Each model has its own strengths and assumptions, so it's important to consider the characteristics of your data and the specific forecasting goals you have in mind.

For guidance on time series analysis, and how to use a time series model, you can refer to [this link](#).