***Data Cleaning***

Missing Value Handling :

*# set seed for reproducibility*

np.random.seed(0)

isnull().sum()

remove missing value from row

.dropna()

remove missing value from coloumn

.dropna(axis=1)

Fill the null values with 0

fillna(0)

*# replace all NA's the value that comes directly after it in the same column,*

*# then replace all the remaining na's with 0*

subset\_nfl\_data.fillna(method='bfill', axis=0).fillna(0)

Scale And Normalization:

* in **scaling**, you're changing the *range* of your data, while
* in **normalization**, you're changing the *shape of the distribution* of your data.

Normalization

.boxcox()

Scaling

minmax\_scaling()

Parsing Dates :

most common are %d for day, %m for month, %y for a two-digit year and %Y for a four digit year.

Some examples:

* 1/17/07 has the format "%m/%d/%y"
* 17-1-2007 has the format "%d-%m-%Y"
* landslides['date\_parsed'] = pd.to\_datetime(landslides['date'], format="%m/**%d**/%y")
* **What if I run into an error with multiple date formats?** While we're specifying the date format here, sometimes you'll run into an error when there are multiple date formats in a single column. If that happens, you have have pandas try to infer what the right date format should be. You can do that like so:

landslides['date\_parsed'] = pd.to\_datetime(landslides['Date'], infer\_datetime\_format=True)

* **Why don't you always use infer\_datetime\_format = True?** There are two big reasons not to always have pandas guess the time format. The first is that pandas won't always been able to figure out the correct date format, especially if someone has gotten creative with data entry. The second is that it's much slower than specifying the exact format of the dates.

*# get the day of the month from the date\_parsed column*

* day\_of\_month\_landslides = landslides['date\_parsed'].dt.day

Character Encoding :

UTF-8 is **the** standard text encoding. All Python code is in UTF-8 and, ideally, all your data should be as well. It's when things aren't in UTF-8 that you run into trouble.

Inconsistent Data Entry :