Day1 Missing Values

1. Take a first look at the data

```python
# modules we'll use
import pandas as pd
import numpy as np

# read in all our data
nfl_data = pd.read_csv("../input/nflplaybyplay2009to2016/NFL Play by Play 2009-2017 (v4).csv")
sf_permits = pd.read_csv("../input/building-permit-applications-data/Building_Permits.csv")

# set seed for reproducibility
np.random.seed(0)
```

```
/opt/conda/lib/python3.6/site-packages/IPython/core/interactiveshell.py:2698: DtypeWarning: Columns (25,51) have mixed types. Specify dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
/opt/conda/lib/python3.6/site-packages/IPython/core/interactiveshell.py:2698: DtypeWarning: Columns (22,32) have mixed types. Specify dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

#set up modules and read in all our data

```python
# look at a few rows of the nfl_data file. I can see a handful of missing data already!
nfl_data.sample(5)
```

	Date	GameID	Drive	qtr	down	time	TimeUnder	TimeSecs	PlayTimeDiff	SideofField
244485	2014-10-26	2014102607	18	3	1.0	00:39	1	939.0	12.0	TB
115340	2011-11-20	2011112000	22	4	1.0	06:47	7	407.0	44.0	OAK
68357	2010-11-14	2010111401	8	2	NaN	00:23	1	1823.0	0.0	CLE
368377	2017-09-24	2017092405	24	4	1.0	08:48	9	528.0	8.0	CLE
384684	2017-11-05	2017110505	11	2	1.0	09:15	10	2355.0	0.0	DEN

look at a few rows of the nfl_data file. I can see a handful of missing data already!

2. See how many missing data points we have

```python
# get the number of missing data points per column
missing_values_count = nfl_data.isnull().sum()

# look at the # of missing points in the first ten columns
missing_values_count[0:10]
```

```
Date                0
GameID              0
Drive               0
qtr                 0
down            61154
time              224
TimeUnder           0
TimeSecs          224
PlayTimeDiff      444
SideofField       528
dtype: int64
```

how many total missing values do we have?
look at the # of missing points in the first ten columns

```python
# how many total missing values do we have?
total_cells = np.product(nfl_data.shape)
print(nfl_data.shape)
total_missing = missing_values_count.sum()


# percent of data that is missing
(total_missing/total_cells) * 100
```

```
(407688, 102)

24.87214126835169
```

how many total missing values do we have?
percent of data that is missing

3. Figure out why the data is missing

This is the point at which we get into the part of data science that I like to call "data intution", by which I mean "really looking at your data and trying to figure out why it is the way it is and how that will affect your analysis". It can be a frustrating part of data science, especially if you're newer to the field and don't have a lot of experience. For dealing with missing values, you'll need to use your intution to figure out why the value is missing. One of the most important question you can ask yourself to help figure this out is this:

> **Is this value missing becuase it wasn't recorded or becuase it dosen't exist?**

If a value is missing becuase it doens't exist (like the height of the oldest child of someone who doesn't have any children) then it doesn't make sense to try and guess what it might be. These values you probalby do want to keep as NaN. On the other hand, if a value is missing becuase it wasn't recorded, then you can try to guess what it might have been based on the other values in that column and row. (This is called "imputation" and we'll learn how to do it next! :)

Let's work through an example. Looking at the number of missing values in the nfl_data dataframe, I notice that the column `TimesSec` has a lot of missing values in it:

By looking at the documentation, I can see that this column has information on the number of seconds left in the game when the play was made. This means that these values are probably missing because they were not recorded, rather than because they don't exist. So, it would make sense for us to try and guess what they should be rather than just leaving them as NA's.

On the other hand, there are other fields, like `PenalizedTeam` that also have lot of missing fields. In this case, though, the field is missing because if there was no penalty then it doesn't make sense to say *which* team was penalized. For this column, it would make more sense to either leave it empty or to add a third value like "neither" and use that to replace the NA's.

> **Tip:** This is a great place to read over the dataset documentation if you haven't already! If you're working with a dataset that you've gotten from another person, you can also try reaching out to them to get more information.

If you're doing very careful data analysis, this is the point at which you'd look at each column individually to figure out the best strategy for filling those missing values. For the rest of this notebook, we'll cover some "quick and dirty" techniques that can help you with missing values but will probably also end up removing some useful information or adding some noise to your data.

3. Drop missing values

```python
# remove all the rows that contain a missing value
nfl_data.dropna()
```

	Date	GameID	Drive	qtr	down	time	TimeUnder	TimeSecs	PlayTimeDiff	SideofField	...	yacEPA

remove all the rows that contain a missing value but it did remove all rows becuse all rows contant miss value

```python
# remove all columns with at least one missing value
columns_with_na_dropped = nfl_data.dropna(axis=1)
columns_with_na_dropped.head()
```

	Date	GameID	Drive	qtr	TimeUnder	ydstogo	ydsnet	PlayAttempted	Yards.Gained	sp	...	T
0	2009-09-10	2009091000	1	1	15	0	0	1	39	0	...	0
1	2009-09-10	2009091000	1	1	15	10	5	1	5	0	...	0
2	2009-09-10	2009091000	1	1	15	5	2	1	-3	0	...	0
3	2009-09-10	2009091000	1	1	14	8	2	1	0	0	...	0
4	2009-09-10	2009091000	1	1	14	8	2	1	0	0	...	0

remove all columns with at least one missing value

```python
# just how much data did we lose?
print("Columns in original dataset: %d \n" % nfl_data.shape[1])
print("Columns with na's dropped: %d" % columns_with_na_dropped.shape[1])
```

```
Columns in original dataset: 102

Columns with na's dropped: 41
```

just how much data did we lose?

4. Filling in missing values automatically

```python
# get a small subset of the NFL dataset
subset_nfl_data = nfl_data.loc[:, 'EPA':'Season'].head()
subset_nfl_data
```

	EPA	airEPA	yacEPA	Home_WP_pre	Away_WP_pre	Home_WP_post	Away_WP_post	Win_Prob	WPA	airWP
0	2.014474	NaN	NaN	0.485675	0.514325	0.546433	0.453567	0.485675	0.060758	NaN
1	0.077907	-1.068169	1.146076	0.546433	0.453567	0.551088	0.448912	0.546433	0.004655	-0.03
2	-1.402760	NaN	NaN	0.551088	0.448912	0.510793	0.489207	0.551088	-0.040295	NaN
3	-1.712583	3.318841	-5.031425	0.510793	0.489207	0.461217	0.538783	0.510793	-0.049576	0.106
4	2.097796	NaN	NaN	0.461217	0.538783	0.558929	0.441071	0.461217	0.097712	NaN

get a small subset of the NFL dataset

```python
# replace all NA's with 0
subset_nfl_data.fillna(0)
```

	EPA	airEPA	yacEPA	Home_WP_pre	Away_WP_pre	Home_WP_post	Away_WP_post	Win_Prob	WPA	airWP
0	2.014474	0.000000	0.000000	0.485675	0.514325	0.546433	0.453567	0.485675	0.060758	0.000
1	0.077907	-1.068169	1.146076	0.546433	0.453567	0.551088	0.448912	0.546433	0.004655	-0.03
2	-1.402760	0.000000	0.000000	0.551088	0.448912	0.510793	0.489207	0.551088	-0.040295	0.000
3	-1.712583	3.318841	-5.031425	0.510793	0.489207	0.461217	0.538783	0.510793	-0.049576	0.106
4	2.097796	0.000000	0.000000	0.461217	0.538783	0.558929	0.441071	0.461217	0.097712	0.000

replace all NA's with 0

```python
# replace all NA's the value that comes directly after it in the same column,
# then replace all the reamining na's with 0
subset_nfl_data.fillna(method = 'bfill', axis=0).fillna("0")
```

	EPA	airEPA	yacEPA	Home_WP_pre	Away_WP_pre	Home_WP_post	Away_WP_post	Win_Prob	WPA	airWPA
0	2.014474	-1.06817	1.14608	0.485675	0.514325	0.546433	0.453567	0.485675	0.060758	-0.0322
1	0.077907	-1.06817	1.14608	0.546433	0.453567	0.551088	0.448912	0.546433	0.004655	-0.0322
2	-1.402760	3.31884	-5.03142	0.551088	0.448912	0.510793	0.489207	0.551088	-0.040295	0.10666
3	-1.712583	3.31884	-5.03142	0.510793	0.489207	0.461217	0.538783	0.510793	-0.049576	0.10666
4	2.097796	0	0	0.461217	0.538783	0.558929	0.441071	0.461217	0.097712	0

replace all NA's the value that comes directly after it in the same column,
then replace all the reamining na's with 0