remove all the rows that contain a missing value 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

<pre># remove all columns with at least one missing value columns_with_na_dropped = nfl_data.dropna(axis=1) columns_with_na_dropped.head()</pre>												
	Date	GameID	Drive	qtr	TimeUnder	ydstogo	ydsnet	PlayAttempted	Yards.Gained	sp		Т
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

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

just how much data did we lose?

his 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 our analysis". It can be a frustrating part of data science, especially if you're newer to the field and don't nave 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

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

Columns with na's dropped: 41

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, hen 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 otice 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.

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 ome "quick and dirty" techniques that can help you with missing values but will probably also end up oving some useful information or adding some noise to your data.

> # 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 airW 2.014474 NaN NaN 0.485675 0.514325 0.546433 0.453567 0.485675 0.060758 NaN 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 -1.712583 3.318841 -5.031425 0.510793 0.489207 0.461217 0.538783 0.510793 -0.049576 0.106 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

<pre># replace all NA's with 0 subset_nfl_data.fillna(0)</pre>												
	EPA	airEPA	yacEPA	Home_WP_pre	Away_WP_pre	Home_WP_post	Away_WP_post	Win_Prob	WPA	airWF		
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

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 2.014474 -1.06817 1.14608 0.485675 0.514325 0.546433 0.453567 0.485675 0.060758 -0.0322 0.077907 -1.06817 1.14608 0.546433 0.453567 0.551088 0.448912 0.546433 0.004655 -0.0322 -1.402760 3.31884 -5.03142 0.551088 0.448912 0.510793 0.489207 0.551088 -0.040295 0.10666 -1.712583 3.31884 -5.03142 0.510793 0.489207 0.461217 0.538783 0.510793 -0.049576 0.10666 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

3. Drop missing values

3. Figure out why the data is missing

Day1 Missing Values

4. Filling in missing values automatically

1. Take a first look at the data

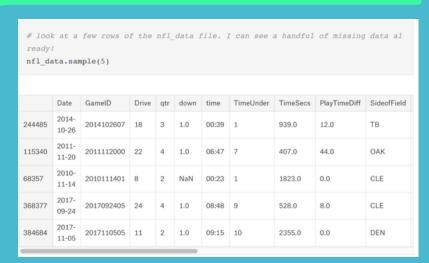
2. See how many missing data

points we have

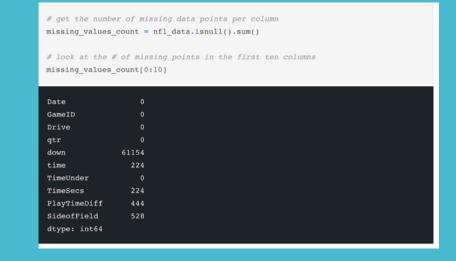
import pandas as pd import numpy as np # read in all our data nfl_data = pd.read_csv("../input/nflplaybyplay2009to2016/NFL Play by Play 2009-20 sf_permits = pd.read_csv("../input/building-permit-applications-data/Building_Per mits.csv") np.random.seed(0) /opt/conda/lib/python3.6/site-packages/IPython/core/interactiveshell.py:2698: Dty peWarning: Columns (25,51) have mixed types. Specify dtype option on import or se $interactivity = interactivity, \ compiler = compiler, \ result = result)$ opt/conda/lib/python3.6/site-packages/IPython/core/interactiveshell.py:2698: Dty eWarning: Columns (22,32) have mixed types. Specify dtype option on import or se

#set up modules and read in all our data

interactivity=interactivity, compiler=compiler, result=result)



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



how many total missing values do we have? # look at the # of missing points

in the first ten columns

total_cells = np.product(nfl_data.shape) print(nfl_data.shape) total missing = missing values count.sum() (total_missing/total_cells) * 100 24.87214126835169

how many total missing values do we have?

percent of data that is missing