

The Investor's DeLorean

See what your investment would have done in the past to make more money in the future!

Meet the Team

Rachel:

Data Wrangler



Sasa:

Our Mascot



Terry:

The Visualizer

Dave:

Data Janitor

Motivation & Summary

Core message

Past performance does not predict future performance...but it can help

DATA TALKS

• We asked these questions...

- If our life expectancies suggest we have 40+ more years to live and we want to invest in property, should we invest in "Bricks & Mortar" in our own Suburbs?
 OR...
- ii. If we can't afford the deposit for a house in our Suburb, should we invest the same amount of money in a select Property Fund/s? AND...
- iii. As it is impossible to precisely predict the future we then asked the question...

"What should we have invested in 10 years ago to have more money now?"

And discovered these answers...

- i. Coding is difficult, APIs can be inconsistent, data is often untidy and unlike it's name suggests, gitting is not fun...at all!
- ii. Property funds are, or at least were a pretty good way to invest money in property if you didn't have enough money to buy a house in the right suburb.

Questions & Data

How has NSW residential housing market performed In the last decade?

Median sale price: Domain.com API_OAuth 2.0

https://developer.domain.com.au/

NSW postcode list:

https://auspost.com.au/business/marketing-and-communications/access-data-and-insights/address-data/postcode-data

How has ASX listed property fund shares performed since 2011?

ASX share price: Yahoo!Finance market data downloader https://pypi.org/project/yfinance/

ASX Real Estate & Property Companies:

https://www.listcorp.com/asx/sectors/real-estate



url_detailed = 'https://api.domain.com.au/ v2/suburbPerformanceStatistics/{}/{}/{}? propertyCategory={}&bedrooms={}& periodSize={}& startingPeriodRelativeToCurrent={}& totalPeriods={}'.format(state,suburb, postcode,propertyCategory,bedrooms, periodSize,startingPeriodRelativeToCurrent, totalPeriods)

to access Domain.com API

Property Market Data Request

numpy to select sample suburbs

#select suburbs starts from 2000 Sydney to eliminate postcodes in our df which are PO BOX

mask1=df['Postcode']>1999

list_suburbs=df[maskl]

We, humans are biased, so let the Algorithm make the choice

list_suburbs=list_suburbs.loc[np.random. choice(list_suburbs.index,100,replace=False)]

list_suburbs=list_suburbs.to_dict()["Postcode"]

list_suburbs

File Handling

def get_property_info(suburb, postcode)

```
res2 = requests.get(url_detailed,headers = auth)
r2=res2.json()
json_object = json.dumps(r2,indent=4)
json.dumps(r2,indent=4)

if "header" in json.loads(json_object).keys():

withopen
("./resources/{}_{}_{}_{})json".format(postcode,suburb),"w")
as outfile:
    outfile.write(json_object)
else:

print('Failure in',suburb, postcode,e,access_token)
```

error/exception handling

```
for suburb,postcode in list_suburbs.items():
    try:
        get_property_info(suburb,postcode)
        print('Success', suburb, postcode)
        final_list_suburbs.append(suburb)

except Exception as e:
    print('Failure in',suburb, postcode,e)
    pass
```

write into&save as .Json



"Think this is bad? You should see the inside of my head."

Median Sold Price and NaNs

■ NSW_pı	rop_data_cleaning.ip:	⊞ data.csv	×					
Delimiter:	, ~							
	year	month	suburb	medianSoldPrice	numberSold	highestSoldPrice	IowestSoldPrice	medianSaleListingPr
122	2010.0	ø.u	OI Levilarus					
222	2016.0	12.0	St Leonards		7.0	5806000.0	1450000.0	
223	2017.0	3.0	St Leonards		9.0	10393000.0	4731000.0	
224	2017.0	6.0	St Leonards	7074000.0	11.0	10192000.0	6325000.0	
225	2017.0	9.0	St Leonards					
226	2017.0	12.0	StLeonards	4387000.0	20.0	7520000.0	2200000.0	
227	2018.0	3.0	St Leonards	5406000.0	16.0	7000000.0	1680000.0	
228	2018.0	6.0	St Leonards		1.0	4900000.0	4900000.0	
229	2018.0	9.0	St Leonards					
230	2018.0	12.0	St Leonards		1.0	6200000.0	6200000.0	
231	2019.0	3.0	St Leonards					
232	2019.0	6.0	St Leonards		1.0	12050000.0	12050000.0	
233	2019.0	9.0	St Leonards					
234	2019.0	12.0	StLeonards					
235	2020.0	3.0	St Leonards					
236	2020.0	6.0	StLeonards					
237	2020.0	9.0	St Leonards					
238	2020.0	12.0	StLeonards					
239	2021.0	3.0	St Leonards		1.0	3050000.0	3050000.0	
240	2021.0	6.0	St Leonards		1.0	33813000.0	33813000.0	
241	2011.0	9.0	Mosman	2080000.0	80.0	6800000.0	760000.0	2925000.0
242	2011.0	12.0	Mosman	2078000.0	96.0	19000000.0	700000.0	2900000.0

Looping to find the suburbs with NaNs and appending those suburbs to a new dataframe...

Then going through the original dataframe and making a new dataframe of suburbs that aren't on the list of suburbs with NaNs...

Then checking we have 40 rows for each suburb on that new dataframe

```
[8]: list sub = []
     for i in range(len(df)):
         if df['medianSoldPrice'][i] == 'NaN':
             list sub.append(df['suburb'][i])
[9]: value_list = list(set(list_sub))
     value list
[9]: ['Blair Athol',
      'Pyangle',
      'Yannergee',
      'Indi',
      'Bowman Farm',
      'Bombo',
      'Canonba'.
      'Pooncarie'.
      'Huntingwood',
      'Maroota',
      'Holbrook',
     bool_val = ~df.suburb.isin(value_list)
     df2 = df[bool val]
     df2['suburb'].value counts()
121: Marsfield
```

Transforming floats in 2 columns into datetime...

_				
	year	month	suburb	medianSoldPrice
	2011.0	9.0	Bathurst	275000.0
	2011.0	12.0	Bathurst	288000.0
	2012.0	3.0	Bathurst	293000.0
	2012.0	6.0	Bathurst	315000.0
	2012.0	9.0	Bathurst	287000.0
		0	Bathurst	310000.0

[41]:	import pandas as pd	0	Bathurst	310000.0
	import numpy as np	0	Batterst	280000.0
	<pre>df = pd.DataFrame(np.random.randint(1,12,size=(100, 2)), columns=list('AB')) df['A']=df['A']+2000</pre>	0	Bathurst	317000.0
	<pre>df['Combined']=df['A'].astype(str)+'-'+df['B'].astype(str)+'-'+str(1) df['Combined_converted']=pd.to_datetime(df['Combined'])</pre>	0	Bathu	335000.0
		0	Bathurst	330000.0
	<pre>df['Quarter'] = pd.PeriodIndex(df['Combined_converted'], freq='Q') df['Recalculated_Date']=pd.PeriodIndex(df['Quarter'], freq='Q').to_timestamp(</pre>		Bathurst	325000.0
	df	1	A SECTION AND ADDRESS OF THE PARTY OF THE PA	

[41]:		Α	В	Combined	Combined_converted	Quarter	Recalculated_Date
	0	2001	8	2001-8-1	2001-08-01	2001Q3	2001-07-01
	1	2007	1	2007-1-1	2007-01-01	2007Q1	2007-01-01
	2	2006	1	2006-1-1	2006-01-01	2006Q1	2006-01-01

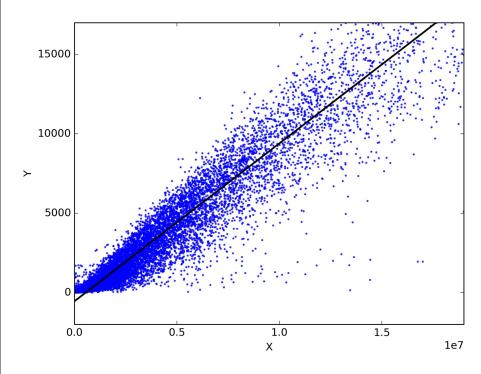
Visualizations...



Pre-Postmortem

Create a model to predict house prices using Python Scikit-learn Package & linear regression models





```
"SCG.AX: Scentre Group",

"CHC.AX: Charter Hall",

"LLC.AX: LendLease Group",

"SGP.AX: Stockland",

"DXS.AX: Dexus",

"MGR.AX: Mirvac",

"LIC.AX: Lifestyle Communities",

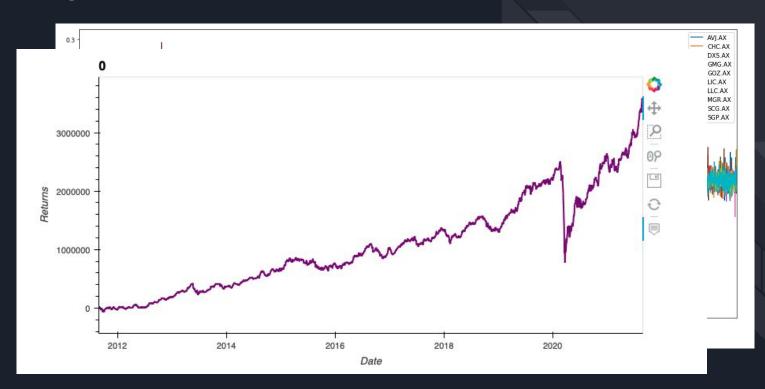
"AVJ.AX: Avjennings",

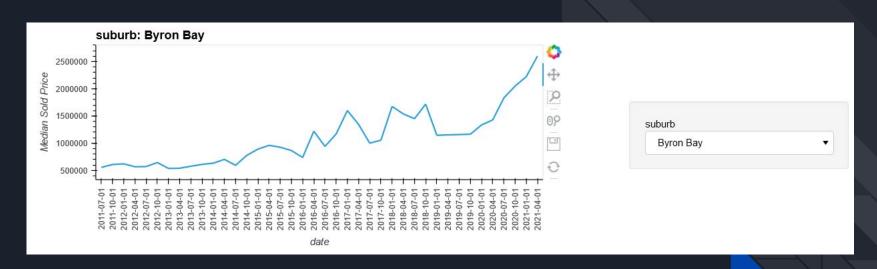
"GOZ.AX: Growthpoint Properties",

"GMG.AX: Goodman Group"
```

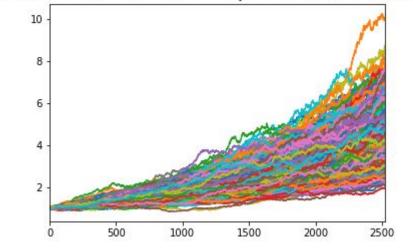
```
Ticker = ["SCG.AX","CHC.AX","LLC.AX","SGP.AX","DXS.AX","MGR.AX","LIC.AX","AVJ.AX","GOZ.AX","GMG.AX"]

Ticker_AX=[]
Property_Fund_Co_Summary=[]
for ticker in Ticker:
    ticker_AX=yf.Ticker(ticker)
# print(ticker_AX)
    ticker_info=ticker_AX.info['longBusinessSummary']
    Property_Fund_Co_Summary.append(ticker_info)
# print(ticker_info)
```

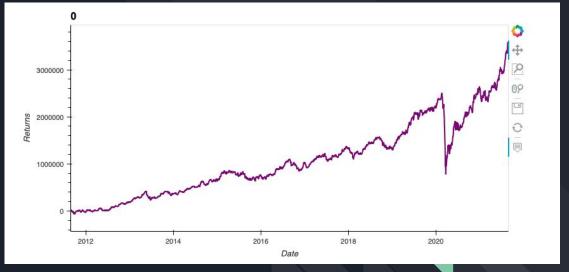


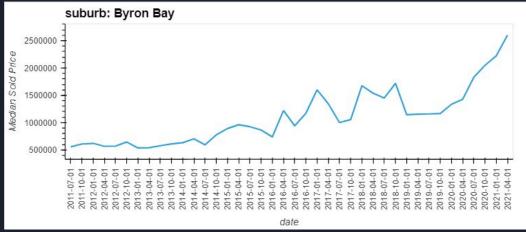


ations of Cumulative Portfolio Return Trajectories Over the Next 2520 Tr



Conclusion







Questions?

