Part 2: A New Way to Approach a Traditional JP Morgan Read Through:

Driving our American Express (AXP) model with JP Morgan (JPM) results, with code

Intro text: In this series we are using the Candas Data Science Library to evaluate the revenue impact of reported Q4 2021 JPM financials results on companies in related financial sub sectors who have yet to report Q4 earnings.

The following is an example of the use of Canalyst Fundamental Data using Candas, not intended to be a recommendation of action.

The Python Jupyter Notebooks supporting this analysis can be found at our Candas Github Repository**.**

JPMorgan Chase & Co. (NYSE: JPM) reported Q4 2021 earnings on January 14. Historically, it has been the first money center bank to report earnings. Given that it is a money center bank, JPM touches on many subsectors within financials such as:

- Card Processing (V MA GPN)
- Credit Cards (COF DFS AXP)
- Capital Markets (GS MS)
- Securities Servicing (STT BK)

Consensus opinion holds that these stocks should trade on their corresponding line items from JPM.

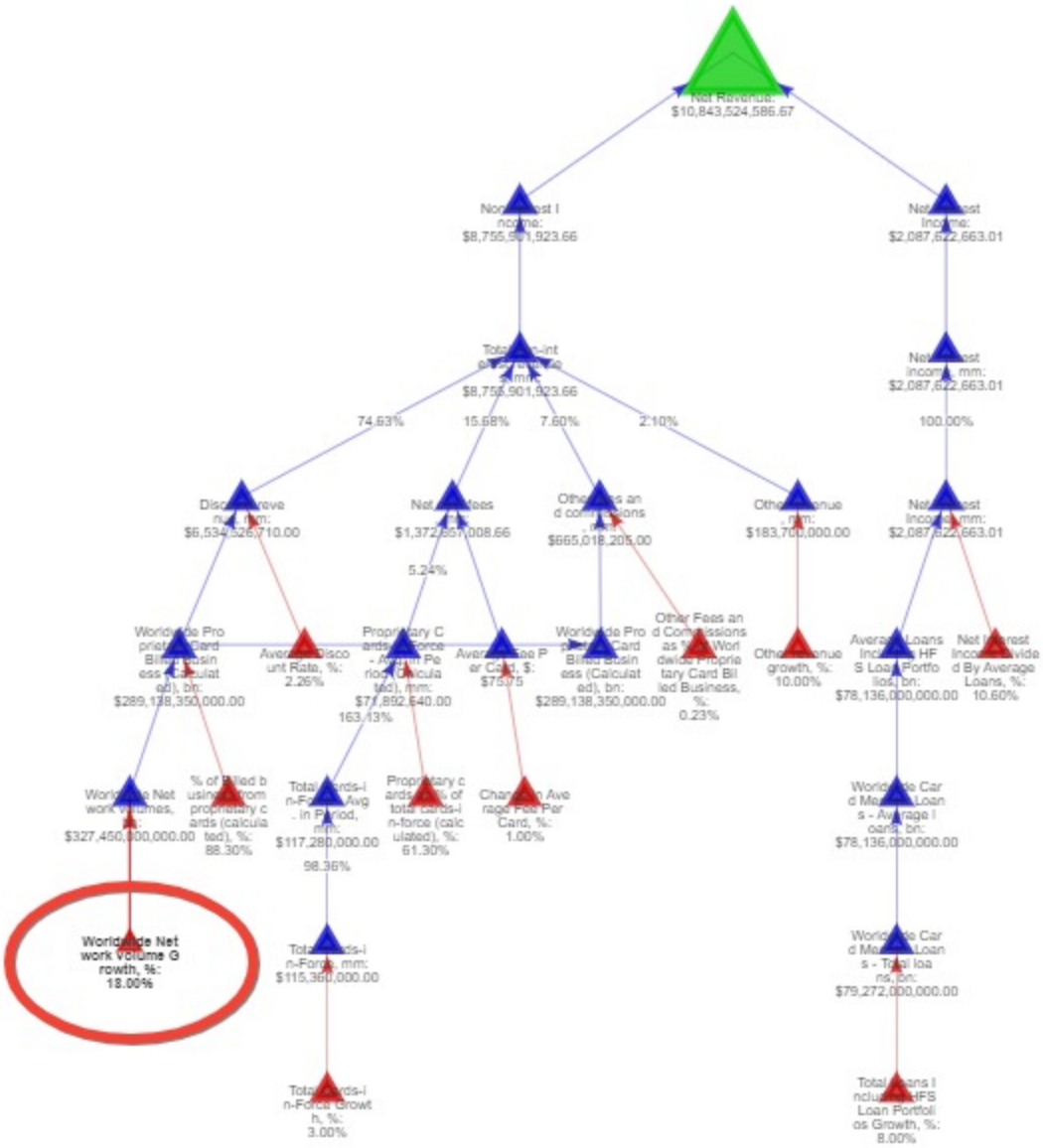
In this brief report we outline some conclusions for American Express (AXP) based on JPM Q4 results.

TLDR Conclusion:

1. **JPM results caused us to revise our AXP revenue estimates upwards by 6%**
2. **We are now slightly ahead of the Street**
3. **AXP tends to beat earnings, however AXP stock tends not to show subsequent alpha, on average over 52 observations**
4. **It is likely therefore that buy side estimates tend to sit higher than Street estimates for this company, on average over time**
5. **One surprising caveat is that based on our key KPI analysis, AXP historically tends to trade more on Cards in Force than Volume**

The six steps we took to get to our conclusion:

1. **Use the AXP ModelMap from our Candas library to find a likely KPI pair candidate, in this case "WorldWide Network Volume Growth"**



2. **Create a dataset from each (JPM and AXP) model:**

Create searchable KPI sets for each company

```
df_jpm_kpi = jpm_model_set.model_frame(mrq=True)[['time_series_name','time_series_description','unit_type']]
df_target_kpi = target_model_set.model_frame(mrq=True)[['time_series_name','time_series_description','unit_type']]
```

3. **Find the model-specific "time series name" for WorldWide Network Volume in AXP using the kpi_match() function:**

Use the kpi_match() function to find "Volume" in the target company KPI list

We find "WorldWide Network Volume Growth %" as the top match

```
kpi_match(df_target_kpi,"volume","",10)
```

	time_series_description	time_series_name
1	Worldwide Network volume Growth, %	MO_GA_TPV
2	Worldwide Card Member Receivables - GCP Net loss ratio as a % of charge volume, %	z_FYXPQQ0114_MO_OS_WorldwideCardMemberReceivablesGCP90dayspastbillingasaoftotal

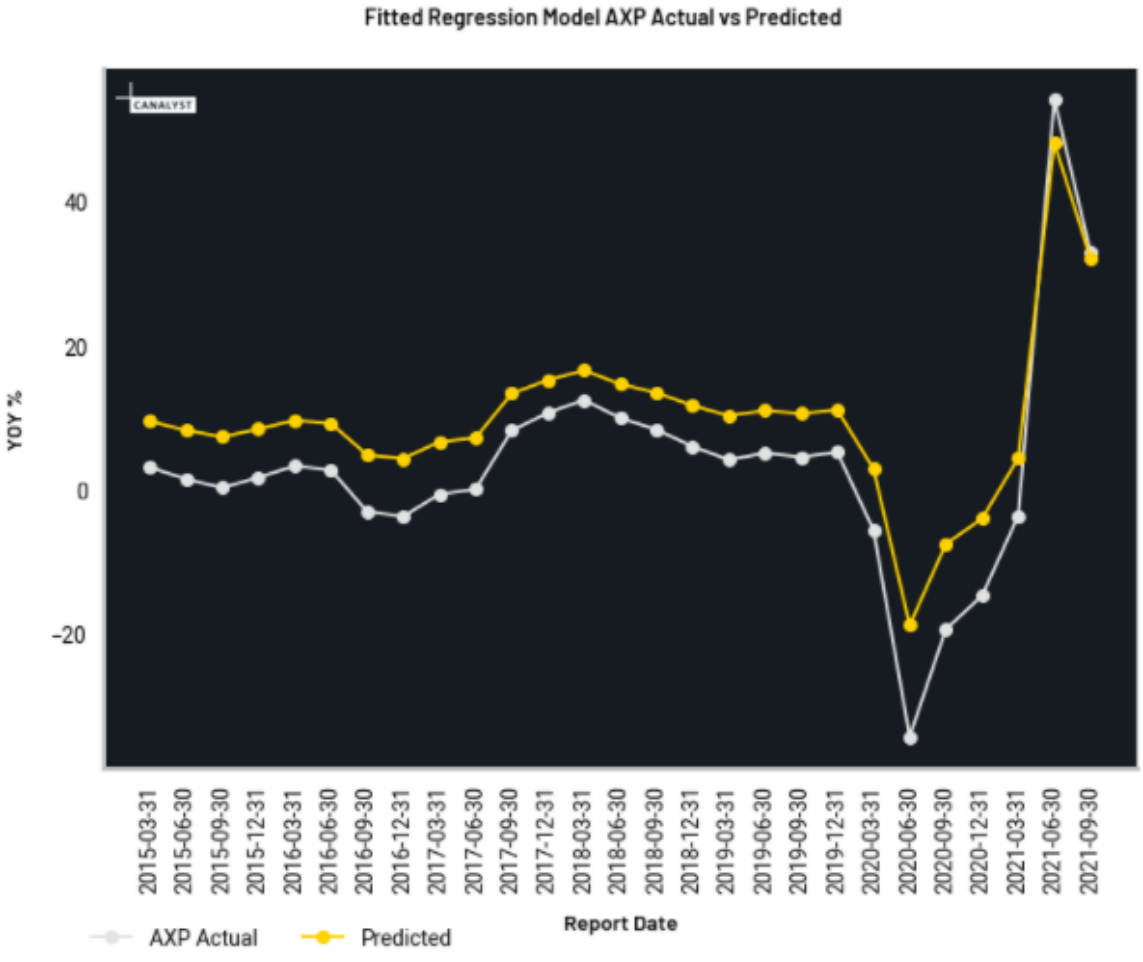
4. **Having created a dataframe with our paired series (using Credit Card Sales Volume from JPM as highlighted in our previous report titled "Observations on Traditional JP Morgan Read Through"):**

Now we can run a regression against our two time series

```
import statsmodels.formula.api as smf
mod = smf.ols(formula='value_x ~ value_y', data=df)
res = mod.fit()
res.summary().tables[0]
```

OLS Regression Results

Dep. Variable:	value_x	**R-squared:**	0.914



5. **RSQ of .91 and a good fit - we then use that regression formula to forecast AXP based on JPM for Q4**

And we can use our regression formula to forecast the target Q4 volume number

```
intercept = res.params[0]
beta = res.params[1]
new_jpm_value = 29
y = beta*new_jpm_value + intercept
print(y)
```

```
29.09129265591725
```

6. **Now we re-drive the AXP model with this new number for Q4**

We are going to set that value to our forecasted number of 29

```
df_params = target_model_set.forecast_frame("MO_GA_TPV",
                            n_periods=-1,
                            function_name='value',
                            function_value=29)

# How much does revenue move?
return_series = "MO_RIS_REV"

dicts_output=target_model_set.fit(df_params,return_series)
```

```
AXP US scenario_id_url: https://mds.canalyst.com/api/equity-model-series/FYXPQQ01
```

Conclusion: Our revenues bumped 6% higher, and took us slightly over the Street for the quarter.

This creates a 6% bump in revenues versus the Canalyst model

```
dicts_output['AXP US'].head(1)
```

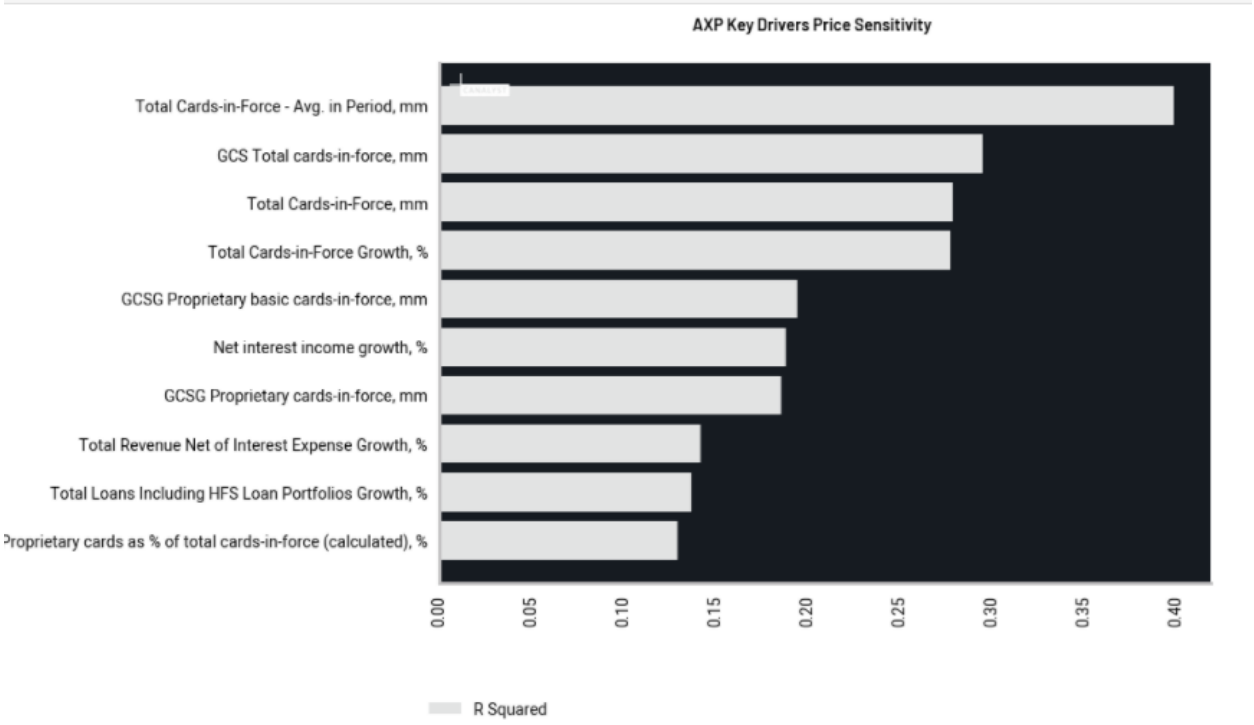
	ticker	period_name	time_series_name	time_series_description	default	scenario	diff
0	AXP US	Q4-2021	MO_RIS_REV	Net Revenue	10843524586.67124	11514668604.1712	1.06189

Which takes us to 11.5bn revenues vs the street of 11.47

```
cdr.get_revenue_estimates(yahoo_ticker)
```

	Revenue Estimate	Current Qtr. (Dec 2021)	Next Qtr. (Mar 2022)	Current Year (2021)	Next Year (2022)
0	No. of Analysts	21	12	25	25
1	Avg. Estimate	11.47B	10.95B	41.75B	47.7B
2	Low Estimate	10.96B	10.2B	41.19B	45.21B
3	High Estimate	11.86B	11.41B	42.27B	49.77B
4	Year Ago Sales	9.35B	9.17B	36.09B	41.75B
5	Sales Growth (year/est)	22.60%	19.40%	15.70%	14.20%

Caveat: AXP stock appears to trade more on Cards in Force than network volume, though the R Squared is probably too low to be conclusive.


AXP Key Drivers Price Sensitivity
Total Cards-in-Force - Avg. in Period, mm
GCS Total cards-in-force, mm
Total Cards-in-Force, mm
Total Cards-in-Force Growth, %
GCSG Proprietary basic cards-in-force, mm
Net interest income growth, %
GCSG Proprietary cards-in-force, mm
Total Revenue Net of Interest Expense Growth, %
Total Loans Including HFS Loan Portfolios Growth, %
Proprietary cards as % of total cards-in-force (calculated), %
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
R Squared