Open in Colab

Factors

- Value
- Momentum
- Quality
    - Profitability
    - Low accruals
    - Low asset growth
    - Low default probability
- Volatility (low vol and/or low idiosyncratic vol)
- Liquidity (high volume)

Data

- closeadj, closeunadj, volume from sep_weekly
- marketcap, pb from weekly
- netinc, equity, assets, ncfo from sf1 where dimension="ARQ"

Financial Statement Variables

- Use trailing 4 quarters:
    - netinc, ncfo = sum of prior 4 quarters
    - equity, assets = average of prior 4 quarters
- Variables:
    - roe = netinc / equity
    - accruals = (netinc - ncfo) / equity
    - agr = % change in assets

Create connection

In [1]:

```python
import pandas as pd

from sqlalchemy import create_engine
import pymssql
server = 'fs.rice.edu'
database = 'stocks'
username = 'stocks'
password = '6LAZH1'
string = "mssql+pymssql://" + username + ":" + password + "@" + server + "/"
conn = create_engine(string).connect()
```

Calculate financial ratios and growth rates

Data from SF1

In [2]:
```python
sf1 = pd.read_sql(
    """
    select ticker, datekey, lastupdated, netinc, ncfo, equity, assets
    from sf1
    where dimension='ARQ' and datekey>='2009-01-01' and equity>0 and assets>0
    order by ticker, datekey
    """,
    conn,
    parse_dates=["datekey"]
)
sf1 = sf1.groupby(["ticker", "datekey", "lastupdated"]).last()
sf1 = sf1.droplevel("lastupdated")
sf1 = sf1.reset_index()
```

```
In [3]: for col in ["netinc", "ncfo"]:
            sf1[col] = sf1.groupby("ticker", group_keys=False)[col].apply(lambda x: x
        for col in ["equity", "assets"]:
            sf1[col] = sf1.groupby("ticker", group_keys=False)[col].apply(lambda x: x

        sf1["roe"] = sf1.netinc / sf1.equity
        sf1["accruals"] = (sf1.netinc - sf1.ncfo) / sf1.equity
        sf1["agr"] = sf1.groupby("ticker", group_keys=False)["assets"].pct_change()
        sf1 = sf1[["ticker", "datekey", "roe", "accruals", "agr"]].dropna()
```

Returns, momentum, volatility, closeunadj

Data from sep_weekly

```
In [4]: sep_weekly = pd.read_sql(
            """
            select ticker, date, closeadj, closeunadj, lastupdated
            from sep_weekly
            where date>='2010-01-01'
            order by ticker, date, lastupdated
            """,
            conn,
            parse_dates=["date"]
        )
        sep_weekly = sep_weekly.groupby(["ticker", "date", "lastupdated"]).last()
        sep_weekly = sep_weekly.droplevel("lastupdated")
```

In [5]:

```
sep_weekly["ret"] = sep_weekly.groupby("ticker", group_keys=False).closeadj.p
sep_weekly["annual"] = sep_weekly.groupby("ticker", group_keys=False).closead
sep_weekly["monthly"] = sep_weekly.groupby("ticker", group_keys=False).closea
sep_weekly["mom"] = sep_weekly.groupby("ticker", group_keys=False).apply(
    lambda d: (1+d.annual)/(1+d.monthly) - 1
)
sep_weekly["vol"] = sep_weekly.groupby("ticker", group_keys=False).ret.apply(
sep_weekly = sep_weekly[["ret", "mom", "vol", "closeunadj"]].dropna().reset_i
```

Get marketcap and pb

Data from weekly

```
In [6]: weekly = pd.read_sql(
            """
            select ticker, date, marketcap, pb, lastupdated
            from weekly
            where date>='2010-01-01' and marketcap>0 and pb>0
            order by ticker, date, lastupdated
            """,
            conn,
            parse_dates=["date"]
        )
        weekly = weekly.groupby(["ticker", "date", "lastupdated"]).last()
        weekly = weekly.droplevel("lastupdated")
        weekly = weekly.dropna().reset_index()
```

Merge data

In [7]:

```python
df = weekly.merge(sep_weekly, on=["ticker", "date"], how="inner")
for col in ["pb", "mom", "vol", "marketcap", "closeunadj"]:
    df[col] = df.groupby("ticker", group_keys=False)[col].shift()
df["year"] = df.date.apply(lambda x: x.isocalendar()[0])
df["week"] = df.date.apply(lambda x: x.isocalendar()[1])
sf1["year"] = sf1.datekey.apply(lambda x: x.isocalendar()[0])
sf1["week"] = sf1.datekey.apply(lambda x: x.isocalendar()[1])
df = df.merge(sf1, on=["ticker", "year", "week"])
```

Filter to small caps and exclude penny stocks

```
In [8]: df = df[df.closeunadj>5]
        df["rnk"] = df.groupby("date").marketcap.rank(
            ascending=False,
            method="first"
        )
        df = df[(df.rnk>1000) & (df.rnk<=3000)]
        df = df.reset_index().set_index(["ticker", "date"])
```

```
In [10]: df = df.drop(columns=["marketcap", "closeunadj", "year", "week", "datekey", "
         df.head()
```

Out[10]:

		index	pb	ret	mom	vol	roe	accruals
ticker	**date**							
AACH	**2017-05-05**	91	1.0	-0.008357	-0.577655	0.127541	-0.010791	-0.012126
	2017-08-04	92	1.0	-0.062780	-0.687003	0.056854	-0.027568	-0.040935
	2018-11-09	97	1.0	-0.188073	-0.095122	0.066100	-0.220314	-0.125066
AAIC	**2015-05-08**	143	0.8	-0.075194	0.085155	0.024263	-0.148604	-0.297240
	2015-11-06	145	0.6	-0.050866	-0.381870	0.036616	-0.216424	-0.402808