FX Leverage Carry-Trade

Groupname: Blue Horseshoe

Background

The project seeks to implement the well-known carry trade in Forex markets, which involves borrowing money in a low-yielding currency and investing it for a given period of time in a high-yielding currency. In our case, we consider ourselves as investors possessing JPY, and investing this in currencies including the GBP, AUD and USD. Since currency markets can be volatile, signals usually don't last for more than 2 or 3 months at a stretch - hence, we focus our efforts on rebalancing at tenors (time periods) of 1 week, 1 month and 2 months. We note that forward data for the AUDJPY for 1 week was not available, so this criteria was omitted from our analysis.

Goals

The two sets of decisions as investors that we have to make would be the choice of currency, as well as the duration of the investment.

Methods & Features

We tackle both decision by the use of the interest rate parity signal, which is

$$\frac{F(1+r_f)}{S} - (1 + r_d)$$

where F is the forward rate, S the spot rate and r_f and r_d the foreign and domestic currencies respectively (note the values of spot and forward rates are computed as CCYJPY, with CCY being the foreign currency).

As r_f varies across foreign currency and time periods, and r_d solely across time periods, we determine the currency and time period that generates the highest possible signal to invest in. Once this is decided, we 1) convert the JPY into the currency 2) invest in the existing LIBOR/risk-free rate of the currency 3) convert the currency back into JPY at the prevailing rates at the end of the cycle 4) Repeat the process by determining the best signals.

In case none of the signals are positive, we choose to invest our equity overnight in an money-market fund offering a 0 % interest rate. The reason for this is that interest rates have been historically low in Japan (< 1%), with rates becoming negative since 2016. Our only alternative is to invest at the JPY LIBOR, which offers potentially negative interest rates. Most money managers and banks offer overnight rates at a slightly higher rate than the JPY

LIBOR, which we assume to be 0 %. Furthermore, since we only consider a minimum tenor of 1 week, keeping our equity locked up over this period could potentially lead to us missing a 'good' signal.

Another important feature of our strategy is the use of leverage. Most carry trades involve leverage, since the FX markets can often have long spells (regimes) of sideways movement with minimal returns. Leverage serves to amplify these returns, but can also contribute to magnified losses which could potentially wipe out the entire equity leading to bankruptcy.

Transaction Costs

We also consider the effect of transaction costs. One of the biggest source is from the bid-ask spread of Forex. Such transaction costs are inevitable since the spread is determined by Forex market maker. One way to minimize the spread is to implement the trade when spread is narrow. However, this may lead to lost trade opportunity costs. After trade-off, we decided to directly take the spread into account, that is, we buy the FX at ask price and sell them at bid price. Then, the formula for PNL is

$$L * ((1 + r_f) * \frac{f_{bid} - f_{ask}}{f_{ask}} + r_f - r_d) + r_d$$

where L is the leverage, r_f is the foreign interest rate, r_d is the domestic interest rate and f_{bid} and f_{ask} are the bid and ask prices of the spot rate (Malz, 2019).

The final result of our strategies show that we are still able to make profit even after considering the spread.

In [17]:
```python
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import carry_trade as ct
import datetime as dt

from date_function_v2 import holiday_adjust

matplotlib.rcParams['figure.figsize'] = (16, 9)
font = {'family': 'normal',
        'size': 20}

matplotlib.rc('font', **font)
```

1. Load data

In [2]:
```python
data_path_new = 'Data/to_send.csv'
final_data_new = pd.read_csv(data_path_new)
```

In [3]:
```python
final_data_new = final_data_new.set_index('Date')
final_data_new.index = pd.to_datetime(final_data_new.index)
```

In [4]:
```python
#final_data_new.iloc[324:]
```

2. Implement Strategy

We implement the strategy for two broad scenarios, pre- and post-crisis. The pre-crisis scenario involves trading from September 2007 onwards, while the post-crisis scenario avoids some of the major volatility spikes and fluctuations observed in 2008 and invests from 2009 onwards. The scenarios assume a fairly conservative leverage ratio of 2.

In [5]:
```python
# if you want to test single currency or single trading period, ju
fx_list = ['USD', 'AUD', 'GBP']
period_list = [7, 30, 60]
```

In [6]:
```python
results_pre_crisis = ct.algo_loop(final_data_new, fx_list, period_
results_post_crisis = ct.algo_loop(final_data_new.iloc[324:], fx_l
```

```
2019-12-08 20:30:21:784214: Beginning Carry-Trade Strategy run
2019-12-08 20:30:36:394428: Algo run complete.
2019-12-08 20:30:36:394996: Beginning Carry-Trade Strategy run
2019-12-08 20:30:50:010569: Algo run complete.
```

In [7]:
```python
results_pre_crisis.to_csv('Results/results_pre_crisis.csv')
results_post_crisis.to_csv('Results/results_post_crisis.csv')
```

In [8]:
```
results_post_crisis
```
Out[8]:

Date	Signal	FX_name	Period	Foreign_IR	Domestic_IR	FX_Rate	Equity	Asset Pos	Ur
2009-01-02	0.000484649	GBP	2M	0.02505	0.0075625	133.563	10000	20000	
2009-01-05	0.000484649	GBP	2M	0.02445	0.0073875	137.348	10000	20000	
2009-01-06	0.000484649	GBP	2M	0.0240875	0.0071875	139.69	10000	20000	
2009-01-07	0.000484649	GBP	2M	0.023675	0.007075	139.85	10000	20000	
2009-01-08	0.000484649	GBP	2M	0.0230375	0.0069375	138.773	10000	20000	
...	...	...	...	...	...	...	...	...	
2019-11-05	5.13588e-05	AUD	2M	0.0092	-0.000965	75.247	16811.8	33623.7	
2019-11-06	5.13588e-05	AUD	2M	0.0093	-0.000935	75.024	16811.8	33623.7	
2019-11-07	5.13588e-05	AUD	2M	0.009183	-0.0010383	75.385	16811.8	33623.7	
2019-11-08	5.13588e-05	AUD	2M	0.009216	-0.00109	74.938	16811.8	33623.7	
2019-11-11	5.13588e-05	AUD	2M	0.0091	-0.001135	74.704	16811.8	33623.7	

2713 rows × 11 columns

In [9]:
```
print('Cumulative Return in the post-crisis scenario:', results_po
print('Cumulative Return in the pre-crisis scenario:', results_pre
print('Max Drawdown between trades in the post-crisis scenario:',
print('Max Drawdown between trades in the pre-crisis scenario:',
```
```
Cumulative Return in the post-crisis scenario: 68.11848552276105 %
Cumulative Return in the pre-crisis scenario: -35.85947561379189 %
Max Drawdown between trades in the post-crisis scenario: -38.31904787
844942 %
Max Drawdown between trades in the pre-crisis scenario: -72.471131144
26324 %
```

In [10]:
```
# if you want to test single currency or single trading period, ju
usd_list = ['USD']
aud_list = ['AUD']
gbp_list = ['GBP']
```

```
In [11]: results_usd = ct.algo_loop(final_data_new.iloc[324:], usd_list, pe
         results_aud = ct.algo_loop(final_data_new.iloc[324:], aud_list, pe
         results_gbp = ct.algo_loop(final_data_new.iloc[324:], gbp_list, pe
```

```
2019-12-08 20:31:08:265519: Beginning Carry-Trade Strategy run
2019-12-08 20:31:19:734091: Algo run complete.
2019-12-08 20:31:19:734486: Beginning Carry-Trade Strategy run
2019-12-08 20:31:27:125301: Algo run complete.
2019-12-08 20:31:27:125769: Beginning Carry-Trade Strategy run
2019-12-08 20:31:35:604022: Algo run complete.
```

```
In [12]: print('Cumulative Return, USD only:', results_usd['Real_Return'][-
```

```
Cumulative Return, USD only: 1.5332769349171382 %
```

```
In [13]: print('Cumulative Return, AUD only:', results_aud['Real_Return'][-
```

```
Cumulative Return, AUD only: 38.83647339032503 %
```

```
In [14]: print('Cumulative Return, GBP only:', results_gbp['Real_Return'][-
```

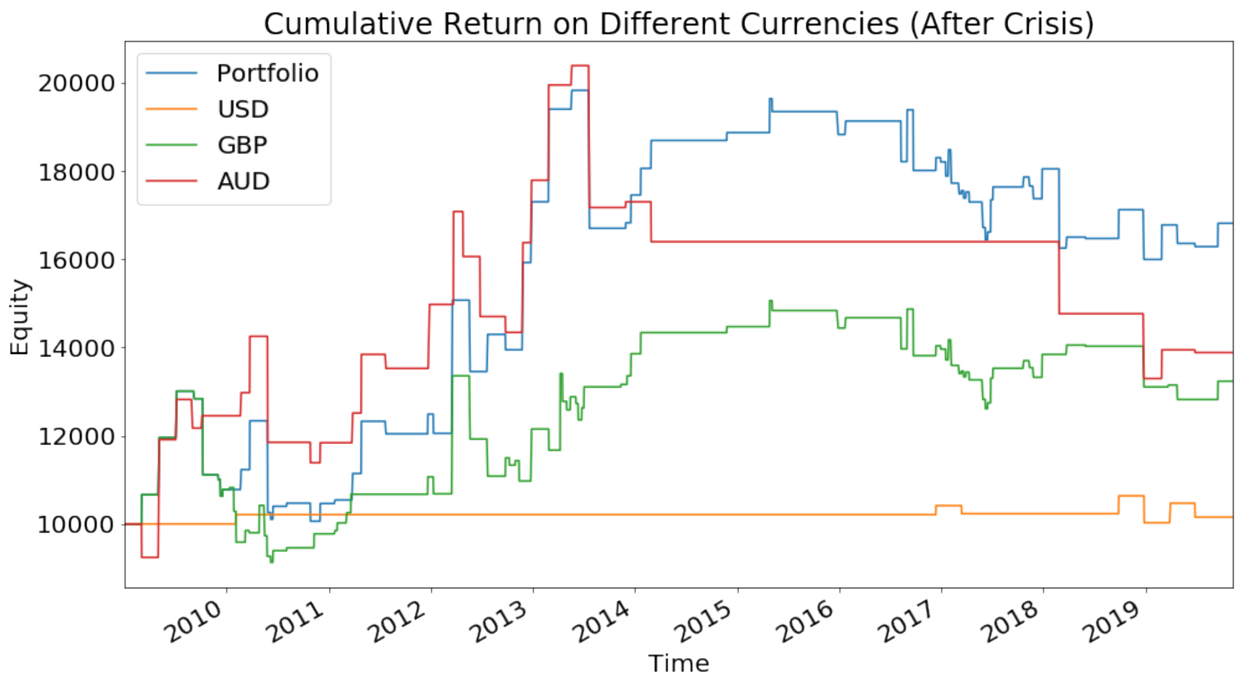
```
Cumulative Return, GBP only: 32.34347226895811 %
```

3. Analysis

3.1 After Crisis

3.1.1 Realized Return Time Series

In [19]:
```
results_post_crisis['Equity'].plot(label='Portfolio')
results_usd['Equity'].plot(label='USD')
results_gbp['Equity'].plot(label='GBP')
results_aud['Equity'].plot(label='AUD')
plt.legend()
plt.xlabel('Time')
plt.ylabel('Equity')
plt.title('Cumulative Return on Different Currencies (After Crisis
plt.show()
```

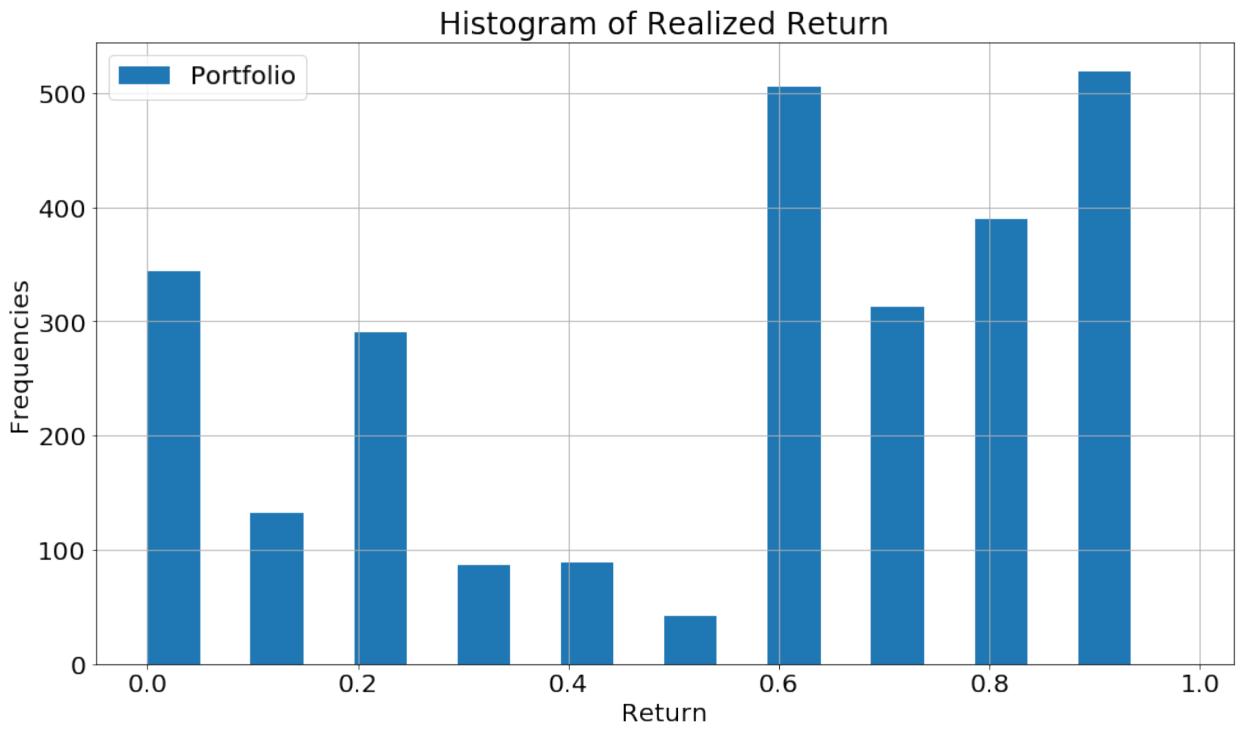


As you can see, the diversified portfolio tends to outperform but closely track the AUD portfolio, which makes sense due to the high spread between the Japanese and Australian interest rates

3.1.2 Realized Return Histogram

In [20]:

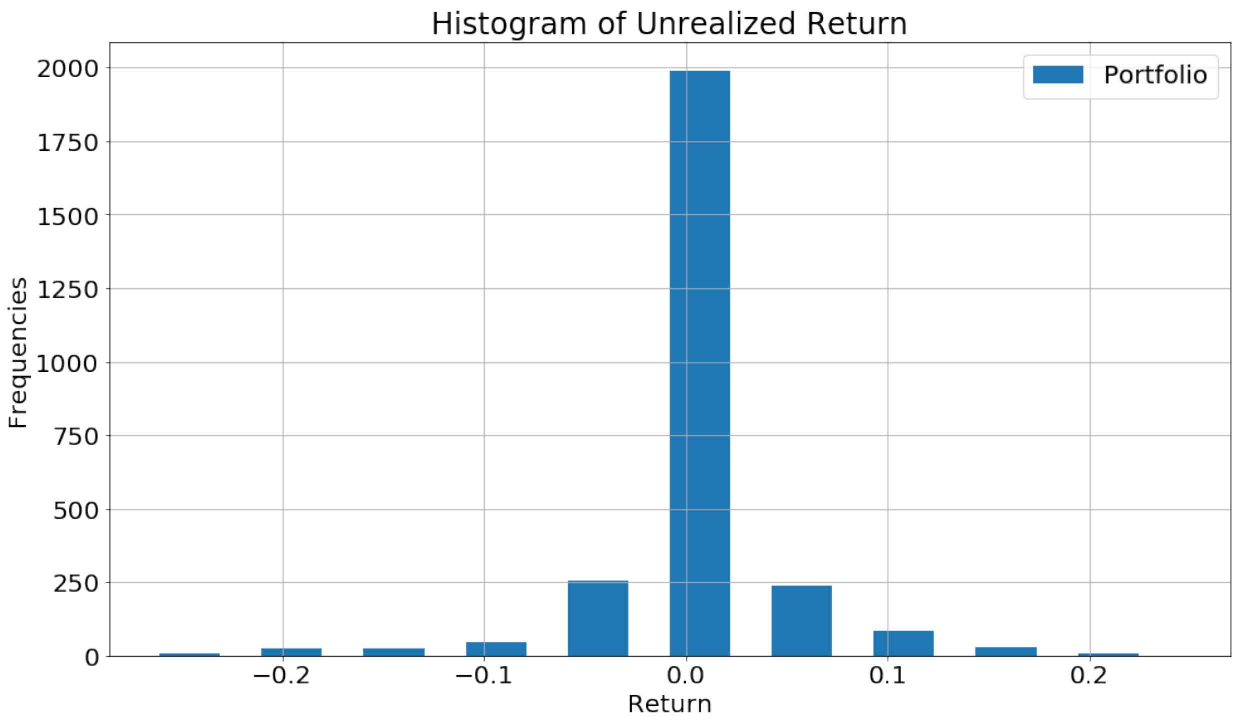
```
results_post_crisis['Real_Return'].hist(width=0.05, label='Portfol
plt.legend()
plt.xlabel('Return')
plt.ylabel('Frequencies')
plt.title('Histogram of Realized Return')
plt.show()
```



Realized return tends to be bimodal in our case, which largely has to do with our fluctuations in return varying slightly in the short-term, but overall trending upwards. This means that there is a period where we hover closer to 0-0.2, but then after a period of realtively solid positive returns we shift over for the most part to fluctuating returns in the 0.6-0.9 range.

3.1.3 Unrealized Return Histogram

In [21]:
```
results_post_crisis['Unreal_Return'].hist(width=0.03, label='Port
plt.legend()
plt.xlabel('Return')
plt.ylabel('Frequencies')
plt.title('Histogram of Unrealized Return')
plt.show()
```



Unlike realized returns, unrealized returns is distributed highly centered around a return of near 0, but slightly skews more positive than negative. this makes sense since for the most part we had modest daily returns that over time added up to having a positive realized in the long run.

3.1.4 Value at Risk

The Value at Risk, or VaR, is an estimate of how much the portfolio can lose over a given time period (one day in our case) at a certain confidence level. For example, a VaR of 10 % at a 95 % confidence level would mean that we are 95 % certain (or that 95 times out of a 100) that a portfolio could sustain losses of no more than 10 % during that day. The higher the confidence level, the greater the loss estimate and hence the greater the VaR. There is both a parametric method as well as a historical method of computing VaR - we use the historical method since we have a sufficient number of data points for a fairly accurate estimation. The methodology involves

- sorting returns from smallest to largest
- choosing the quantile of returns that corresponds to (1 - confidence level)

In [22]:
```
return_post_crisis = results_post_crisis['Unreal_Return'].sort_val
print('VaR at 90% confidence level:', return_post_crisis.quantile(
print('VaR at 95% confidence level:', return_post_crisis.quantile(
print('VaR at 99% confidence level:', return_post_crisis.quantile(
```

```
VaR at 90% confidence level: 1.7924809982412095 %
VaR at 95% confidence level: 4.625538554631749 %
VaR at 99% confidence level: 16.263490527294692 %
```

3.1.5 Sharpe Ratio and Volatility

We measure the daily and annualized sharpe ratios using our unrealized returns. Our opportunity cost r_f is assumed to be 0, hence the formula for Sharpe ratio is given by $\frac{\mu}{\sigma}$, where μ and σ are the mean and standard deviation of the unrealized returns respectively.

In [23]:
```
sharpe_post_crisis =  results_post_crisis['Unreal_Return'].mean()/
print('Daily Sharpe Ratio:', sharpe_post_crisis)
print('Annual Sharpe Ratio:', sharpe_post_crisis * np.sqrt(251))
```

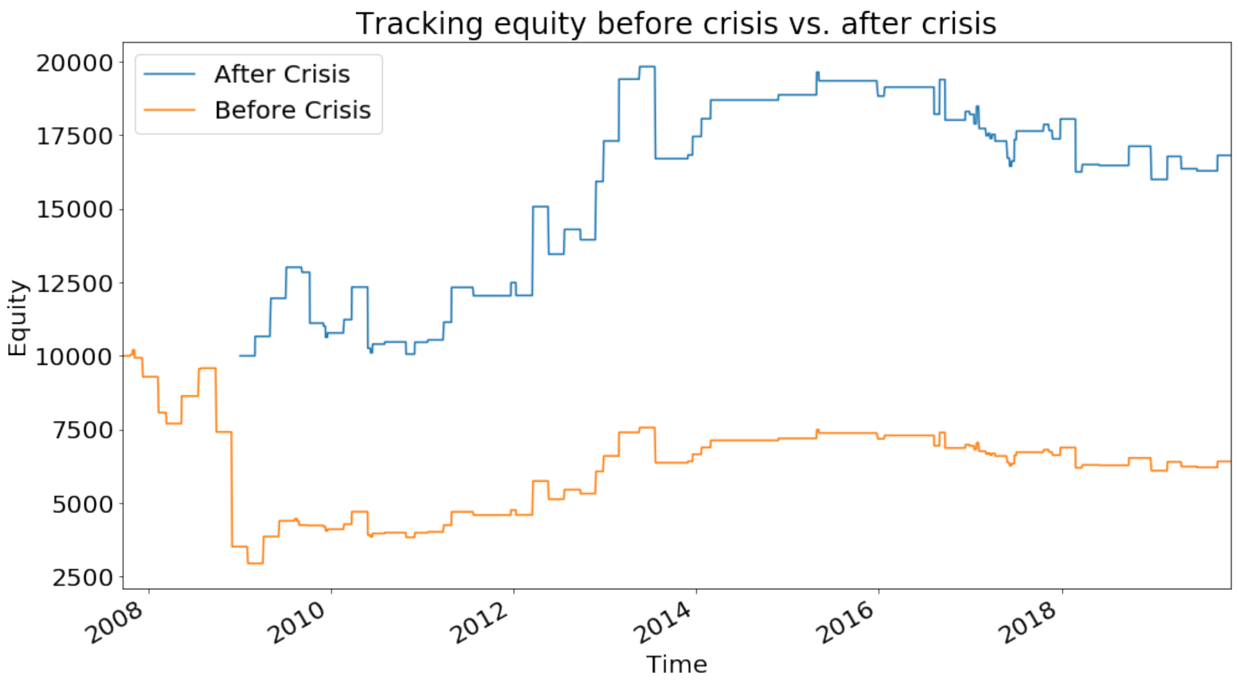
```
Daily Sharpe Ratio: 0.15823949102082854
Annual Sharpe Ratio: 2.5069850151429405
```

In [24]:
```
volatility_post_crisis = results_post_crisis['Unreal_Return'].std(
print('Volatility, after crisis (daily):', volatility_post_crisis*
```

```
Volatility, after crisis (daily): 4.571939757280838 %
```

3.2 Before Crisis vs. After Crisis

In [25]:
```
results_post_crisis['Equity'].plot(label='After Crisis')
results_pre_crisis['Equity'].plot(label='Before Crisis')
plt.legend()
plt.xlabel('Time')
plt.ylabel('Equity')
plt.title('Tracking equity before crisis vs. after crisis')
plt.show()
```



3.2.1 Value at Risk

In [26]:
```
return_post_crisis = results_post_crisis['Unreal_Return'].sort_val
print('VaR at 90%, after crisis:', return_post_crisis.quantile(0.1
print('VaR at 95%, after crisis:', return_post_crisis.quantile(0.0
print('VaR at 99%, after crisis:', return_post_crisis.quantile(0.0
```

```
VaR at 90%, after crisis: 1.7924809982412095 %
VaR at 95%, after crisis: 4.625538554631749 %
VaR at 99%, after crisis: 16.263490527294692 %
```

In [27]:
```
return_pre_crisis = results_pre_crisis['Unreal_Return'].sort_value
print('VaR at 90%, before crisis:', return_pre_crisis.quantile(0.1
print('VaR at 95%, before crisis:', return_pre_crisis.quantile(0.0
print('VaR at 99%, before crisis:', return_pre_crisis.quantile(0.0
```

```
VaR at 90%, before crisis: 3.720340479002022 %
VaR at 95%, before crisis: 9.568580250763162 %
VaR at 99%, before crisis: 23.524271427766017 %
```

3.2.2 Sharpe Ratio and Volatility

In [28]:

```
sharpe_pre_crisis =  results_pre_crisis['Unreal_Return'].mean()/re
print('Daily Sharpe Ratio after crisis:', sharpe_post_crisis)
print('Annual Sharpe Ratio after crisis:', sharpe_post_crisis * np
print('Daily Sharpe Ratio before crisis:', sharpe_pre_crisis)
print('Annual Sharpe Ratio before crisis:', sharpe_pre_crisis * np
```

```
Daily Sharpe Ratio after crisis: 0.15823949102082854
Annual Sharpe Ratio after crisis: 2.5069850151429405
Daily Sharpe Ratio before crisis: 0.011846622873341955
Annual Sharpe Ratio before crisis: 0.18768580353692282
```

In [29]:

```
volatility_pre_crisis = results_pre_crisis['Real_Return'].std()
print('Volatility after crisis (daily):', volatility_post_crisis *
print('Volatility before crisis (daily):', volatility_pre_crisis *
```

```
Volatility after crisis (daily): 4.571939757280838 %
Volatility before crisis (daily): 15.16022133265221 %
```

These results intuitively make sense - the volatility is higher and Sharpe ratio lower for the period starting pre-crisis compared to that of the post crisis scenario.

3.3 Leverage Analysis

In this section, we explore the effect of leverage on our portfolio. As most carry trades assume some leverage to realize meaningful returns, we use a conservative leverage of 2 for most of our project. We now explore how modifying leverage can affect our returns both pre, and post crisis.

The formula for returns (on equity) in the carry trade with leverage is given by $L[(1+r_f)\frac{\delta S}{S} + r_f - r_d] + r_d$, where L is the leverage, r_f is the foreign interest rate, r_d is the domestic interest rate and S is the spot rate (Malz, 2019). The leverage itself is defined as $\frac{Assets}{Equity}$, where $Assets = Equity + Debt$is the total amount involved in the trade. Every trade in a scenario is assumed to have the same leverage, with the amount borrowed being $(Leverage - 1) * Equity$ We explore the pre and post-crisis scenarios independently to determine the influence the highly volatile FX market swings had on portfolios with varying leverages.

3.3.1 After Crisis

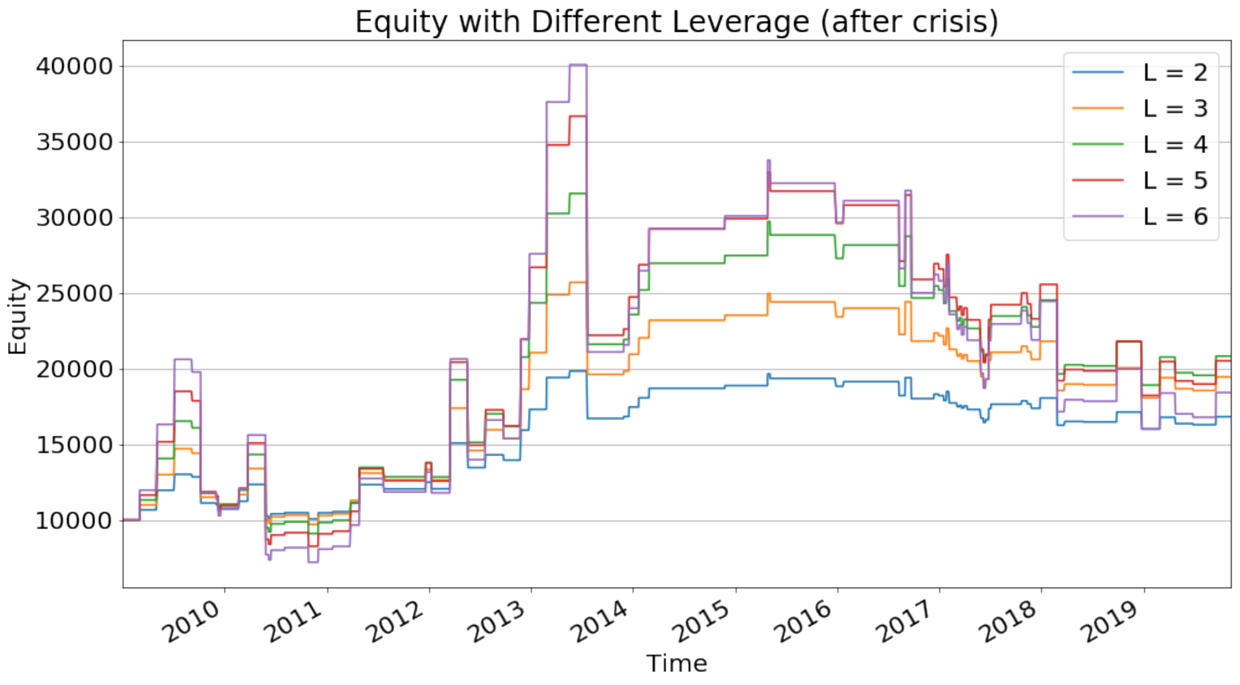
In [30]:

```
results_post_crisis_l3 = ct.algo_loop(final_data_new.iloc[324:], f
results_post_crisis_l4 = ct.algo_loop(final_data_new.iloc[324:], f
results_post_crisis_l5 = ct.algo_loop(final_data_new.iloc[324:], f
results_post_crisis_l6 = ct.algo_loop(final_data_new.iloc[324:], f
```

```
2019-12-08 20:33:26:420803: Beginning Carry-Trade Strategy run
2019-12-08 20:33:40:362806: Algo run complete.
2019-12-08 20:33:40:363255: Beginning Carry-Trade Strategy run
2019-12-08 20:33:54:910750: Algo run complete.
2019-12-08 20:33:54:911217: Beginning Carry-Trade Strategy run
2019-12-08 20:34:10:171447: Algo run complete.
2019-12-08 20:34:10:171929: Beginning Carry-Trade Strategy run
2019-12-08 20:34:24:671753: Algo run complete.
```

In [31]:

```
results_post_crisis['Equity'].plot(label='L = 2')
results_post_crisis_l3['Equity'].plot(label='L = 3')
results_post_crisis_l4['Equity'].plot(label='L = 4')
results_post_crisis_l5['Equity'].plot(label='L = 5')
results_post_crisis_l6['Equity'].plot(label='L = 6')
plt.grid(True, axis='y')
plt.legend(loc='upper right')
plt.xlabel('Time')
plt.ylabel('Equity')
plt.title('Equity with Different Leverage (after crisis)')
plt.show()
plt.savefig('Results/Real_Return_leverages_post_crisis.jpg')
```



```
<Figure size 1152x648 with 0 Axes>
```

This illustrates the point that due to the higher upside volatility of increased leverage, when the markets tend to perform well (i.e. no massive fiscal crisis, or other market shocks), higher leverage tends to have higher returns. However, it does not strictly hold for longer periods of time (> 10 years), where idiosyncratic shocks can still lead to higher losses due to the leverage (the double-edged sword phenomenon). In our case, an optimal leverage of 4 has the best returns, while a leverage of 2 has the highest annualized Sharpe Ratio.

In [32]:
```
sharpe_post_crisis =  results_post_crisis['Unreal_Return'].mean()/
sharpe_post_crisis_l3 =  results_post_crisis_l3['Unreal_Return'].m
sharpe_post_crisis_l4 =  results_post_crisis_l4['Unreal_Return'].m
sharpe_post_crisis_l5 =  results_post_crisis_l5['Unreal_Return'].m
sharpe_post_crisis_l6 =  results_post_crisis_l6['Unreal_Return'].m


print('Leverage = 2, annualized Sharpe Ratio:', sharpe_post_crisis
print('Leverage = 3, annualized Sharpe Ratio:', sharpe_post_crisis
print('Leverage = 4, annualized Sharpe Ratio:', sharpe_post_crisis
print('Leverage = 5, annualized Sharpe Ratio:', sharpe_post_crisis
print('Leverage = 6, annualized Sharpe Ratio:', sharpe_post_crisis

```

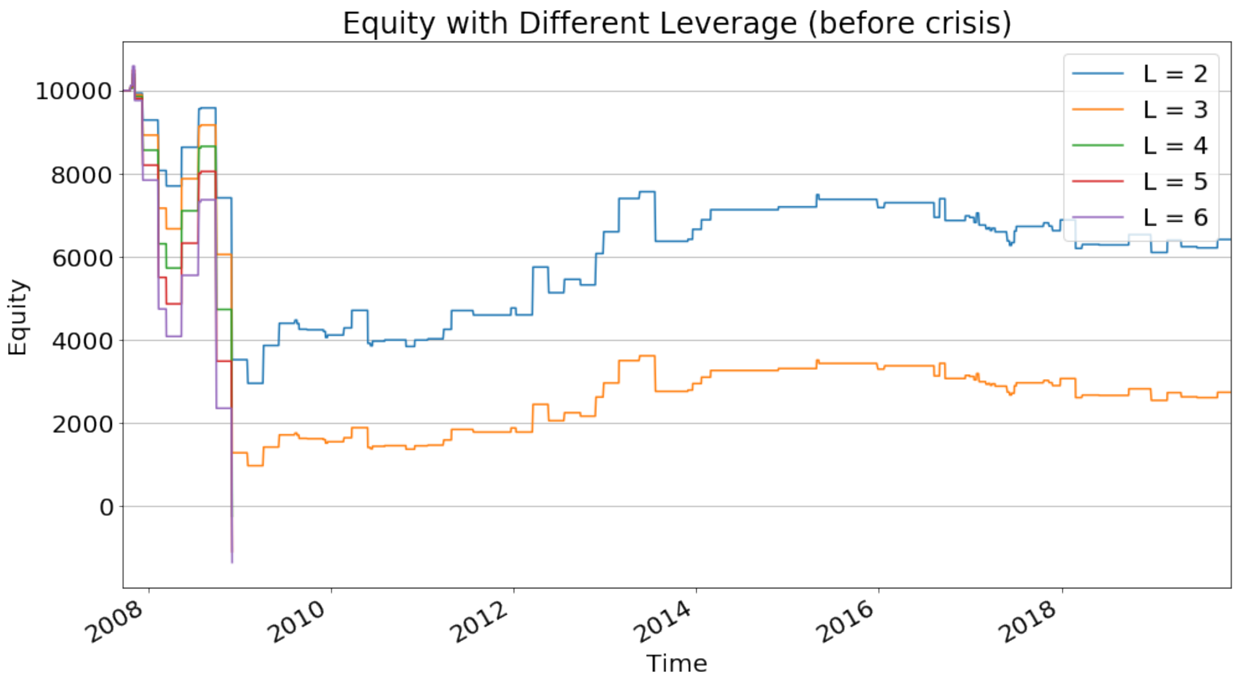
```
Leverage = 2, annualized Sharpe Ratio: 2.5069850151429405
Leverage = 3, annualized Sharpe Ratio: 2.4963635317613964
Leverage = 4, annualized Sharpe Ratio: 2.4910487631126403
Leverage = 5, annualized Sharpe Ratio: 2.4878586178373836
Leverage = 6, annualized Sharpe Ratio: 2.485731320338369
```

3.3.2 Before Crisis

In [33]:
```
results_pre_crisis_l3 = ct.algo_loop(final_data_new, fx_list, peri
results_pre_crisis_l4 = ct.algo_loop(final_data_new, fx_list, peri
results_pre_crisis_l5 = ct.algo_loop(final_data_new, fx_list, peri
results_pre_crisis_l6 = ct.algo_loop(final_data_new, fx_list, peri
```

```
2019-12-08 20:36:17:050566: Beginning Carry-Trade Strategy run
2019-12-08 20:36:32:537874: Algo run complete.
2019-12-08 20:36:32:538052: Beginning Carry-Trade Strategy run
2019-12-08 20:36:33:260349: Algo run complete.
2019-12-08 20:36:33:260529: Beginning Carry-Trade Strategy run
2019-12-08 20:36:33:894650: Algo run complete.
2019-12-08 20:36:33:895093: Beginning Carry-Trade Strategy run
2019-12-08 20:36:34:550835: Algo run complete.
```

In [34]:
```
results_pre_crisis['Equity'].plot(label='L = 2')
results_pre_crisis_l3['Equity'].plot(label='L = 3')
results_pre_crisis_l4['Equity'].plot(label='L = 4')
results_pre_crisis_l5['Equity'].plot(label='L = 5')
results_pre_crisis_l6['Equity'].plot(label='L = 6')
plt.grid(True, axis='y')
plt.legend(loc='upper right')
plt.xlabel('Time')
plt.ylabel('Equity')
plt.title('Equity with Different Leverage (before crisis)')
plt.show()
plt.savefig('Results/Real_Return_leverage_before_crisis.jpg')
```



```
<Figure size 1152x648 with 0 Axes>
```

This illustrates that while higher leverage is beneficiary during times of a good economic climate, when there is a significant crisis such as the forex liquidity crisis a decade ago, higher leverage knocks you out completely and prevents you from ever having the chance to recover your initial losses.

In [35]:
```
sharpe_pre_crisis =  results_pre_crisis['Unreal_Return'].mean()/re
sharpe_pre_crisis_l3 =  results_pre_crisis_l3['Unreal_Return'].mea
sharpe_pre_crisis_l4 =  results_pre_crisis_l4['Unreal_Return'].mea
sharpe_pre_crisis_l5 =  results_pre_crisis_l5['Unreal_Return'].mea
sharpe_pre_crisis_l6 =  results_pre_crisis_l6['Unreal_Return'].mea


print('L2, Sharpe Ratio each year:', sharpe_pre_crisis * np.sqrt(2
print('L3, Sharpe Ratio each year:', sharpe_pre_crisis_l3 * np.sqr
print('L4, Sharpe Ratio each year:', sharpe_pre_crisis_l4 * np.sqr
print('L5, Sharpe Ratio each year:', sharpe_pre_crisis_l5 * np.sqr
print('L6, Sharpe Ratio each year:', sharpe_pre_crisis_l6 * np.sqr
```

```
L2, Sharpe Ratio each year: 0.18768580353692282
L3, Sharpe Ratio each year: 0.17116406192954942
L4, Sharpe Ratio each year: -5.840165927156961
L5, Sharpe Ratio each year: -5.852224515308024
L6, Sharpe Ratio each year: -5.860262097416548
```

4. Reference

- Kent Daniel, Robert J. Hodrick, and Zhongjin Lu (2016) - *The Carry Trade: Risks and Drawdowns* https://www0.gsb.columbia.edu/faculty/rhodrick/papers/DHL_02292016.pdf (https://www0.gsb.columbia.edu/faculty/rhodrick/papers/DHL_02292016.pdf)
- Jeffrey Frankel - *Everything You Always Wanted To Know About The Carry Trade* https://sites.hks.harvard.edu/fs/jfrankel/CarryTradeMilkenInReview.pdf (https://sites.hks.harvard.edu/fs/jfrankel/CarryTradeMilkenInReview.pdf)
- Lukas Menkhoff, Lucio Sarno, Maik Schmeling, Andreas Schrimpf (2011) - *The Risk in Carry Trade* https://voxeu.org/article/risk-carry-trades (https://voxeu.org/article/risk-carry-trades)
- Allan M. Malz, Columbia University Lecture notes - *risk management, public policy, and the financial system: Leverage Risk*
- Dr. David DeRosa (2010) - *Options on Foreign Exchange*, *Interest Parity and Forward, Foreign Exchange*