In [1]:

```
%matplotlib inline

import matplotlib
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import glob
import pandas as pd
import scipy.stats as stats
import pylab as pl
import pymysql

from datetime import timedelta
from sqlalchemy import create_engine
from multiprocessing import Pool, cpu_count

engine = create_engine('mysql+pymysql://root:maxsonic@localhost:3306/gta_data?charset=utf8')
```

In [2]:

```
def applyParallel(dfGrouped, func):
    p = Pool(3)
    ret_list = p.map(func, [group for name, group in dfGrouped])
    p.close()
    p.join()
    return pd.concat(ret_list)
```

In [3]:

```
l = [pd.read_csv(filename, dtype={"Symbol": str}) for filename in glob.glob("./original_data/combine
all_data = pd.concat(l, axis=0)
```

Typesetting math: 0%

In [4]:

```
all_data.drop("UPDATEID", axis=1, inplace=True)
all_data.drop("BUSINESSTIME", axis=1, inplace=True)
all_data.drop("INDUSTRYNAME", axis=1, inplace=True)
all_data.drop("UTSID", axis=1, inplace=True)
all_data.drop("UPDATESTATE", axis=1, inplace=True)
all_data.drop("UPDATETIME", axis=1, inplace=True)
all_data.drop("PreClosePrice", axis=1, inplace=True)
all_data.drop("OpenPrice", axis=1, inplace=True)
all_data.drop("HighPrice", axis=1, inplace=True)
all_data.drop("LowPrice", axis=1, inplace=True)
all_data.drop("Amount", axis=1, inplace=True)
all_data.drop("Distance", axis=1, inplace=True)
all_data.drop("latestTradingDate", axis=1, inplace=True)
all_data.drop("LatestClosePrice", axis=1, inplace=True)
all_data.drop("StateCode", axis=1, inplace=True)
all_data.drop("AvgPrice", axis=1, inplace=True)
all_data.drop("Change", axis=1, inplace=True)
all_data.drop("ChangeRatio", axis=1, inplace=True)
all_data.drop("TotalShare", axis=1, inplace=True)
all_data.drop("CirculatedShare", axis=1, inplace=True)
all_data.drop("TurnoverRate1", axis=1, inplace=True)
all_data.drop("TurnoverRate2", axis=1, inplace=True)
all_data.drop("CirculatedMarketValue", axis=1, inplace=True)
all_data.drop("Amplitude", axis=1, inplace=True)
all_data.drop("RelativeIPOChange", axis=1, inplace=True)
all_data.drop("RelativeIPOChangeRatio", axis=1, inplace=True)
all_data.drop("MinTickSize", axis=1, inplace=True)
all_data.drop("LimitDown", axis=1, inplace=True)
all_data.drop("LimitUp", axis=1, inplace=True)
all_data.drop("CAT_CHANGEDATE", axis=1, inplace=True)
all_data.drop("Volume", axis=1, inplace=True)
all_data.drop("Filling", axis=1, inplace=True)
all_data.drop("SecurityID", axis=1, inplace=True)
```

In [5]:

```
all_data["pb_ratio_adjust"] = 1 / all_data["pb_ratio"]
all_data["MarketValue_adjust"] = 0 - all_data["MarketValue"]
```

In [6]:

```
stock_info = pd.read_sql_query("select * from STK_STOCKINFO", engine)
idx_quotation = pd.read_sql_query("select TRADINGDATE from IDX_MKT_QUOTATION where SYMBOL='000001'",
susp = pd.read_sql_query("select * from STK_SUSPENTIONINFO", engine)
```

```
/home/maxsonic/anaconda2/lib/python2.7/site-packages/pymysql/cursors.py:166: Warnin
g: (1681, u"'@@SESSION.GTID_EXECUTED' is deprecated and will be removed in a future
 release.")
  result = self._query(query)
```

Typesetting math: 0%

In [7]:

```
res = idx_quotation[idx_quotation['TRADINGDATE']=='1990-12-20']
idx_quotation.index.values.size
```

Out[7]:

```
6915
```

In [8]:

```
def remove_stocks(df):
    # no ST and PT
    df = df[~df["ShortName"].str.contains("ST") | ~df["ShortName"].str.contains("PT")]

    # no suspention or resumption in such a range
    #  date - 3 <= supsention_date <= date + 3
    #  date - 3 <= resumption_date <= date + 3
    max_date_idx = idx_quotation.last_valid_index()
    for date in df.TradingDate.unique():
        if idx_quotation[idx_quotation['TRADINGDATE']==date].index.values.size == 0:
            continue
        date_idx = idx_quotation[idx_quotation['TRADINGDATE']==date].index.values[0]
        three_day_before_idx = date_idx - 3 if date_idx - 3 > 0 else 0
        three_day_after_idx = date_idx + 3 if date_idx + 3 <=  max_date_idx else max_date_idx
        susp_list = susp[(susp["SUSPENTIONDATE"] >= idx_quotation.iloc[three_day_before_idx]["TRADIN
        susp_list = susp_list[(susp_list["SUSPENTIONDATE"] <= idx_quotation.iloc[three_day_after_idx
        susp_list = susp_list[(susp_list["RESUMPTIONDATE"] >= idx_quotation.iloc[three_day_before_id
        susp_list = susp_list[(susp_list["RESUMPTIONDATE"] >= idx_quotation.iloc[three_day_after_idx

        df = df[df["Symbol"].isin(susp_list["SYMBOL"]) == False]

    # no stock that is on market for less than 1 year
    for date in df.TradingDate.unique():
        if idx_quotation[idx_quotation['TRADINGDATE']==date].index.values.size == 0:
            continue
        a_year_before_idx = date_idx - 244 if date_idx - 244 > 0 else 0

        not_a_year_old_stock = stock_info[stock_info["LISTEDDATE"] > idx_quotation.iloc[a_year_befor

        df = df[df["Symbol"].isin(not_a_year_old_stock["SYMBOL"]) == False]
    return df

def rank_fun(df):
    df = remove_stocks(df)
    df["pb_ratio_adjust"].rank(ascending=True) / (df.shape[0] + 1)
    df["pb_rank"] = df["pb_ratio_adjust"].rank(ascending=True) / (df.shape[0] + 1)
    df["MarketValue_rank"] = df["MarketValue_adjust"].rank(ascending=True) / (df.shape[0] + 1)
    df["pb_rank_inverse_normal"] = stats.norm.ppf(df["pb_rank"])
    df["MarketValue_rank_inverse_normal"] = stats.norm.ppf(df["MarketValue_rank"])



    df["pb_rank_zscore"] = (df["pb_rank_inverse_normal"] - df["pb_rank_inverse_normal"].mean())/df["
    df["MarketValue_zscore"] = (df["MarketValue_rank_inverse_normal"] - df["MarketValue_rank_inverse

    return df
```

Typesetting math: 0%

In [9]:

```
all_data = applyParallel(all_data.groupby("TradingDate"), rank_fun)
```

```
/home/maxsonic/anaconda2/lib/python2.7/site-packages/scipy/stats/_distn_infrastructu
re.py:1901: RuntimeWarning: invalid value encountered in greater
  cond1 = (0 < q) & (q < 1)
/home/maxsonic/anaconda2/lib/python2.7/site-packages/scipy/stats/_distn_infrastructu
re.py:1901: RuntimeWarning: invalid value encountered in less
  cond1 = (0 < q) & (q < 1)
/home/maxsonic/anaconda2/lib/python2.7/site-packages/scipy/stats/_distn_infrastructu
re.py:1901: RuntimeWarning: invalid value encountered in greater
  cond1 = (0 < q) & (q < 1)
/home/maxsonic/anaconda2/lib/python2.7/site-packages/scipy/stats/_distn_infrastructu
re.py:1901: RuntimeWarning: invalid value encountered in less
  cond1 = (0 < q) & (q < 1)
/home/maxsonic/anaconda2/lib/python2.7/site-packages/scipy/stats/_distn_infrastructu
re.py:1901: RuntimeWarning: invalid value encountered in less
  cond1 = (0 < q) & (q < 1)
/home/maxsonic/anaconda2/lib/python2.7/site-packages/scipy/stats/_distn_infrastructu
re.py:1901: RuntimeWarning: invalid value encountered in greater
  cond1 = (0 < q) & (q < 1)
```

In [ ]:

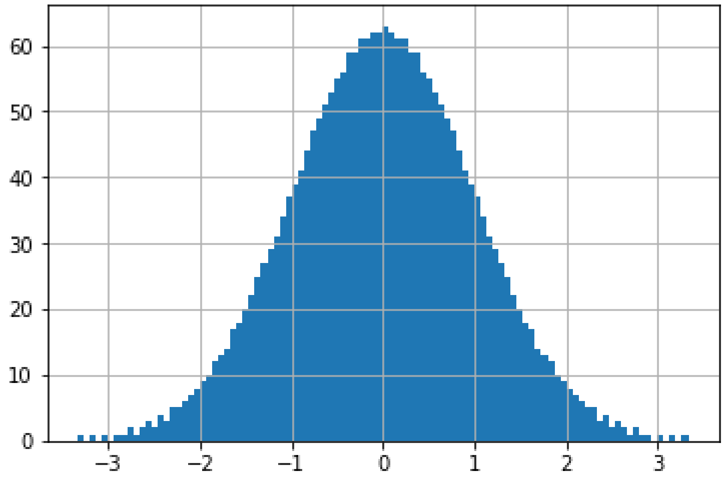
```
date = "2013-06-05"
res = all_data[all_data["TradingDate"] == date]["pb_rank_inverse_normal"]
res.hist(bins=100)
# all_data.loc[all_data["TradingDate"] == date, "pb_rank"] = res
```

In [11]:

```
date = "2013-06-05"
res = all_data[all_data["TradingDate"] == date]["pb_rank_inverse_normal"]
res.hist(bins=100)
# all_data.loc[all_data["TradingDate"] == date, "pb_rank"] = res
```

Out[11]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb3f65a5290>
```



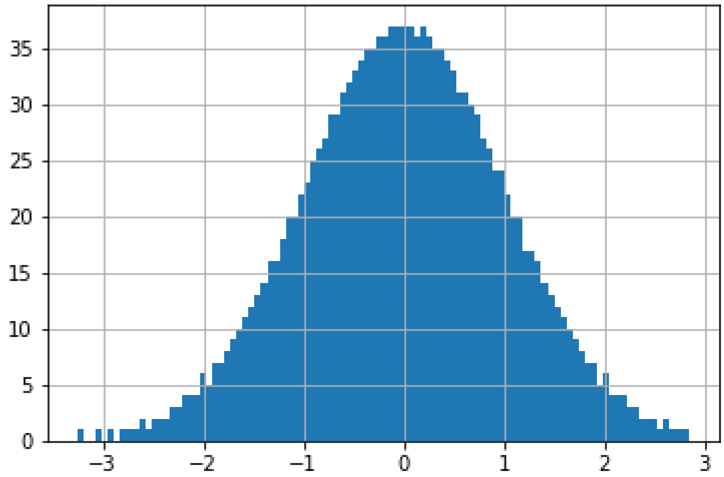
Typesetting math: 0%

In [12]:

```
date = "2009-10-30"
res = all_data[all_data["TradingDate"] == date]["pb_rank_zscore"]
res.hist(bins=100)
```

Out[12]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb3e7e57050>
```



In [13]:

```
all_data_reindex = all_data[["TradingDate", "Symbol", "ShortName", "ClosePrice"]].set_index(["Tradin
forward_return = all_data_reindex.pct_change(5).shift(-5)
forward_return = forward_return.reset_index()
forward_return.head()
```

Out[13]:

	TradingDate	Symbol	ShortName	ClosePrice
0	1990-12-19	600656	ST 博元	-0.251538
1	1990-12-19	600601	方正科技	0.041554
2	1990-12-19	600651	飞乐音响	0.258820
3	1990-12-19	600602	仪电电子	-0.283854
4	1990-12-20	600656	ST 博元	-0.251648

Typesetting math: 0%

In [14]:

```
def return_rank_fun(df):
    df = remove_stocks(df)
    df["close_price_rank"] = df["ClosePrice"].rank(ascending=True) / (df.shape[0] + 1)
    df["close_price_inverse_normal"] = stats.norm.ppf(df["close_price_rank"])

    df["close_price_rank_zscore"] = (df["close_price_inverse_normal"] - df["close_price_inverse_norm

    return df

forward_return = applyParallel(forward_return.groupby("TradingDate"), return_rank_fun)
forward_return.head()
```

```
/home/maxsonic/anaconda2/lib/python2.7/site-packages/scipy/stats/_distn_infrastructu
re.py:1901: RuntimeWarning: invalid value encountered in greater
  cond1 = (0 < q) & (q < 1)
/home/maxsonic/anaconda2/lib/python2.7/site-packages/scipy/stats/_distn_infrastructu
re.py:1901: RuntimeWarning: invalid value encountered in less
  cond1 = (0 < q) & (q < 1)
```

Out[14]:

	TradingDate	Symbol	ShortName	ClosePrice	close_price_rank	close_price_inverse_normal
0	1990-12-19	600656	ST 博元	-0.251538	0.4	-0.253347
1	1990-12-19	600601	方正科技	0.041554	0.6	0.253347
2	1990-12-19	600651	飞乐音响	0.258820	0.8	0.841621
3	1990-12-19	600602	仪电电子	-0.283854	0.2	-0.841621
4	1990-12-20	600656	ST 博元	-0.251648	0.2	-0.841621

Typesetting math: 0%