In [1]:

```
import numpy as np
import pandas as pd
import cdms2, vcs, cdutil, cdtime
from genutil import statistics
from eofs.cdms import Eof
import matplotlib.pyplot as plt
```

In [2]:

```
file = cdms2.open("/Users/bengoldman/HadISST_sst.nc")
```

In [3]:

```
sst = file("sst")
```

```
/opt/anaconda3/envs/cdat81/lib/python3.6/site-packages/cdms2/axis.py:
1685: UserWarning:
Your first bounds[0,0] -180.000000000000000 will be corrected to -180
.000000000000000
Your bounds bounds[-1,1] 180.000000000000000 will be corrected to 180
.000000000000000
  warnings.warn(msg, UserWarning)
```

In [4]:

```
canvas = vcs.init()
```

In [5]:

```
atlantic = sst(lat = (-20,20), long = (360-60, 20))
```

In [6]:

```
pacific = sst(lat = (-5,5), long = (160, 360-90))
```

In [7]:

```
cdutil.setTimeBoundsMonthly(atlantic)
```

In [8]:

```
atlJJA = cdutil.JJA.departures(atlantic)
```

In [9]:

```
solverAtl = Eof(atlJJA)
```

In [10]:

```
atlEof = solverAtl.pcs(npcs = 1)
```

In [11]:

```
cdutil.setTimeBoundsMonthly(pacific)
```

In [12]:

```
pfcDJF = cdutil.DJF.departures(pacific)
```

In [13]:

```
atlEof.info()
```

```
*** Description of Slab pcs ***
id: pcs
shape: (150, 1)
filename:
missing_value: 1e+20
comments:
grid_name: N/A
grid_type: N/A
time_statistic:
long_name: principal_components
units:
tileIndex: None
No grid present.
** Dimension 1 **
   id: time
   Designated a time axis.
   units:  days since 1870-1-1 0:0:0
   Length: 150
   First:  197.0
   Last:   54618.0
   Other axis attributes:
      axis: T
      calendar: gregorian
   Python id:  0x118a0c668
** Dimension 2 **
   id: pc
   Length: 1
   First:  0
   Last:   0
   Other axis attributes:
      long_name: pc_number
   Python id:  0x118a0c518
*** End of description for pcs ***
```

In [14]:

```
pfcAvg = cdutil.averager(pfcDJF, axis="xy" )
```

In [15]:

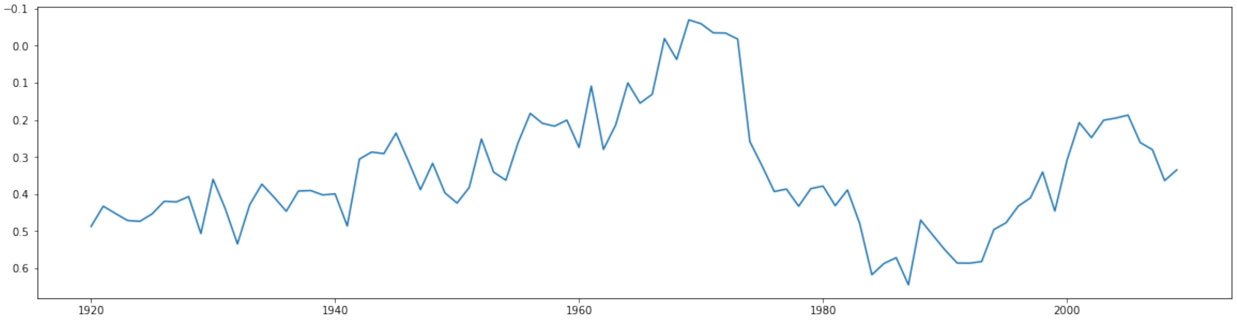
```
coor = []
for i in range(20,150):
    c = statistics.correlation(pfcAvg[i-20:i], atlEof[i-20:i])[0]
    coor.append(c)
```

In [16]:

```
f=plt.figure(figsize=(20,5))
ax = f.subplots()
plt.gca().invert_yaxis()
ax.plot(range(1920,2010), coor[1910-1870:2010-1870])
```

Out[16]:

```
[<matplotlib.lines.Line2D at 0x113e3d240>]
```


−0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
1920
1940
1960
1980
2000