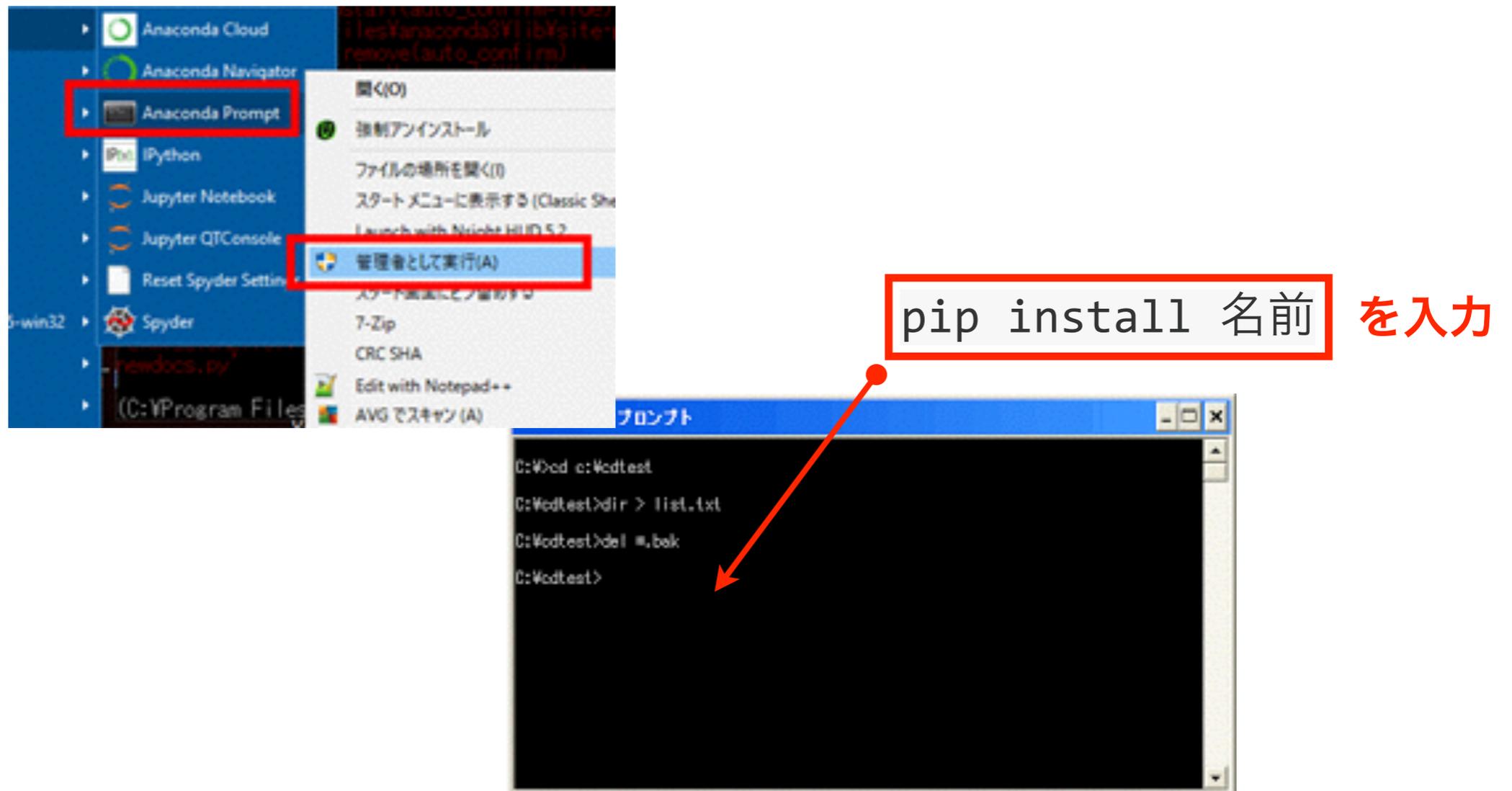


Windowsにおける Pythonの利用の補足

[ポイント1] pipコマンドで外部ライブラリーをインストールする際に、
コマンドプロンプトよりも「アナコンダプロンプト」がベターです。

スタートメニューでAnacondaプロンプト (Anaconda Prompt) を管理者として実行



[ポイント2] pipによる外部ライブラリーをインストールする際に、一部のライブラリーにおいて「Visual C++ (VC)」のコンパイラが必要です。

<http://landinghub.visualstudio.com/visual-cpp-build-tools>

その他のツールとフレームワーク

Visual Studio Team Services CLI (プレビュー) Windows, Linux, Mac から、Visual Studio Team Services のソース コマンド ライン インターフェイスを開きます。コマンド プロンプトまたはスクリプトから、プル要求、ビルド、作業項目などをより直接的に管理します。詳細については、ドキュメントを参照してください。マニュアルのダウンロードやインストール手順については、次のリンクを確認してください - Windows, Linux, Mac。

.NET Framework .NET Framework は、Windows PC、デバイス、サーバーで実行するモバイル、デスクトップ、Web アプリケーションの作成に役立ちます。

Build Tools for Visual Studio 2017 これらの Build Tools を使用すると、Visual Studio IDE を必要とせずに、MSBuild ベースのネイティブおよびマネージ アプリケーションをビルドできます。Visual C++ コンパイラやライブラリ、MFC、ATL、C++/CLI サポートをインストールするオプションもあります。

Visual Studio 2017 の Microsoft Visual C++ 再頒布可能パッケージ Visual Studio 2017 がインストールされていないコンピューター上であっても、このパッケージによって Visual C++ ライブラリのランタイム コンポーネントがインストールされ、そのようなアプリケーションの実行に使用できます。

次のリンクを右クリックしてコピーするかブックマークしてください。
[Visual Studio 2017 の Microsoft Visual C++ 再頒布可能パッケージ](#)

(注) Web画面のかなり下にあります

インストール完了後に pipコマンドを実行してください。(再起動が必要かもしれません)

インストールしています - Visual Studio Build Tools 2017 - 15.3.0

ワークロード 個別のコンポーネント 言語パック

Visual C++ Build Tools Visual C++ ツールセットの機能、ATL、および MFC や C++/CLI などのオプション機能を使用して従来の...

Web 開発ビルド ツール Web アプリケーションのビルドのための MSBuild タスクとターゲット。

.NET Core ビルド ツール .NET Core アプリケーションのビルド用ツール。

場所 C:\Program Files (x86)\Microsoft Visual Studio\2017\BuildTools

実行すると、選択した Visual Studio のエディションのライセンスに同意することになります。また、Visual Studio を使用して他のソフトウェアをダウンロードする機能も提供されます。このソフトウェアは、サードパーティに発する通知または付属するライセンスに記載のとおり、個別にライセンスされています。続行することで、これらのライセンスにも同意することになります。

概要

> MSBuild Tools

Visual C++ Build Tools 含まれるアクセス許可

- ✓ Visual C++ Build Tools のコア機能
- ✓ VC++ 2017 v141 ツールセット (x86,x64)
- ✓ Visual C++ 2017 再頒布可能パッケージの更新プ...

オプション

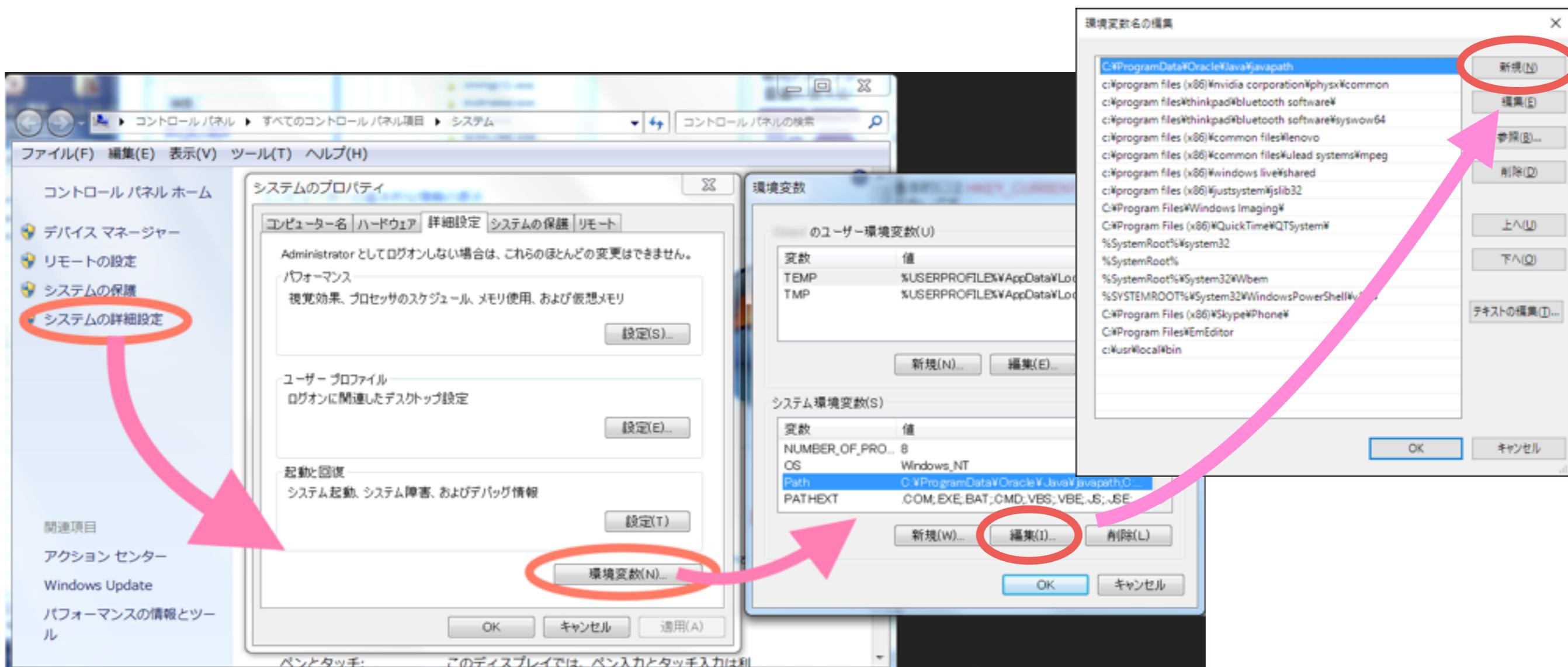
- デスクトップ C++ x86 および x64 用 Windows 1...
- CMake の Visual C++ ツール
- Windows 8.1 SDK と UCRT SDK
- Visual C++ ATL のサポート
- MFC と ATL のサポート (x86 と x64)
- C++/CLI サポート
- Clang/C2 (試験的)
- 標準ライブラリ (のモジュール、pthread等)

合計インストールサイズ: 4.86 GB

インストール

[ポイント3] Rとの連携にて「r = pyper.R()」を実行する前に、
環境変数のPATHに「Rのフォルダ」を追加しておく必要があります。

- ・「コントロールパネル」より以下を辿ってください。
- ・新規のPathとして「C:\Program Files\R\R-3.4.1\bin\x64\」を追加してください。
- ・Rのバージョンが異なる場合や32bit版の場合は、上記のPathが異なります。
- ・C:\Program Files\R\ 以下で「R.exe」があるフォルダを指定してください。

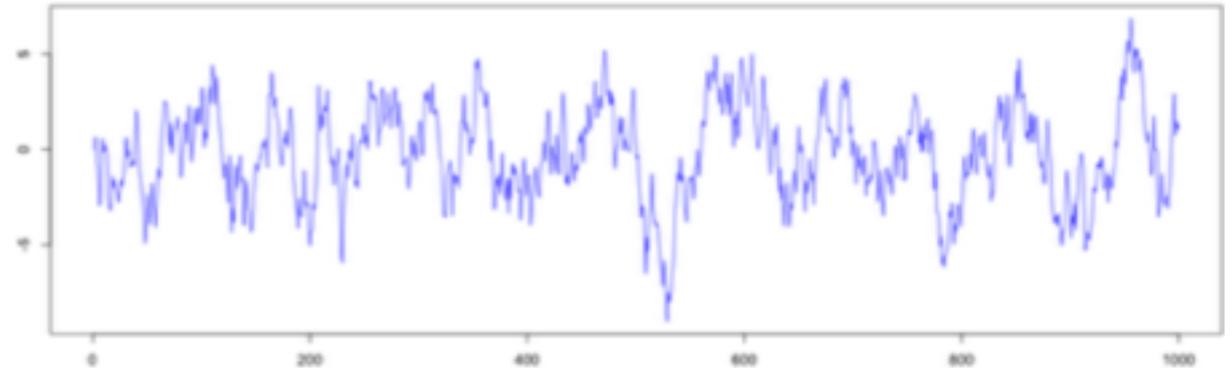


デモ

$$\begin{cases} y(t) = 0.9y(t-1) + e_1(t) \\ x(t) = 0.4x(t-1) + 0.4y(t-1) + e_2(t) \end{cases} \leftarrow \text{変数}x\text{は, }y\text{からの因果を受けている}$$

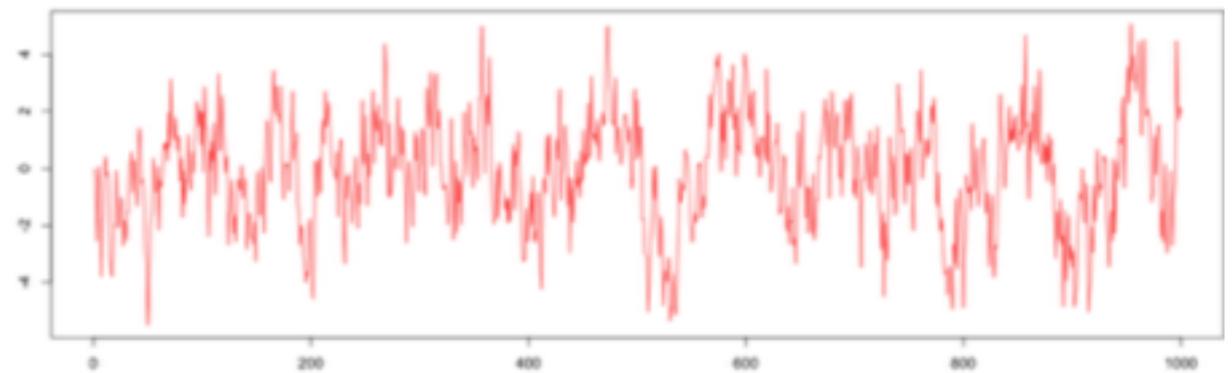
```
y = np.zeros(1000)
for t in range(1, 1000):
    y[t] = 0.9*y[t-1] + np.random.randn(1)

plt.plot(y)
```



```
x = np.zeros(1000)
for t in range(1, 1000):
    x[t] = 0.4*x[t-1] + 0.4*y[t-1] + np.random.randn(1)

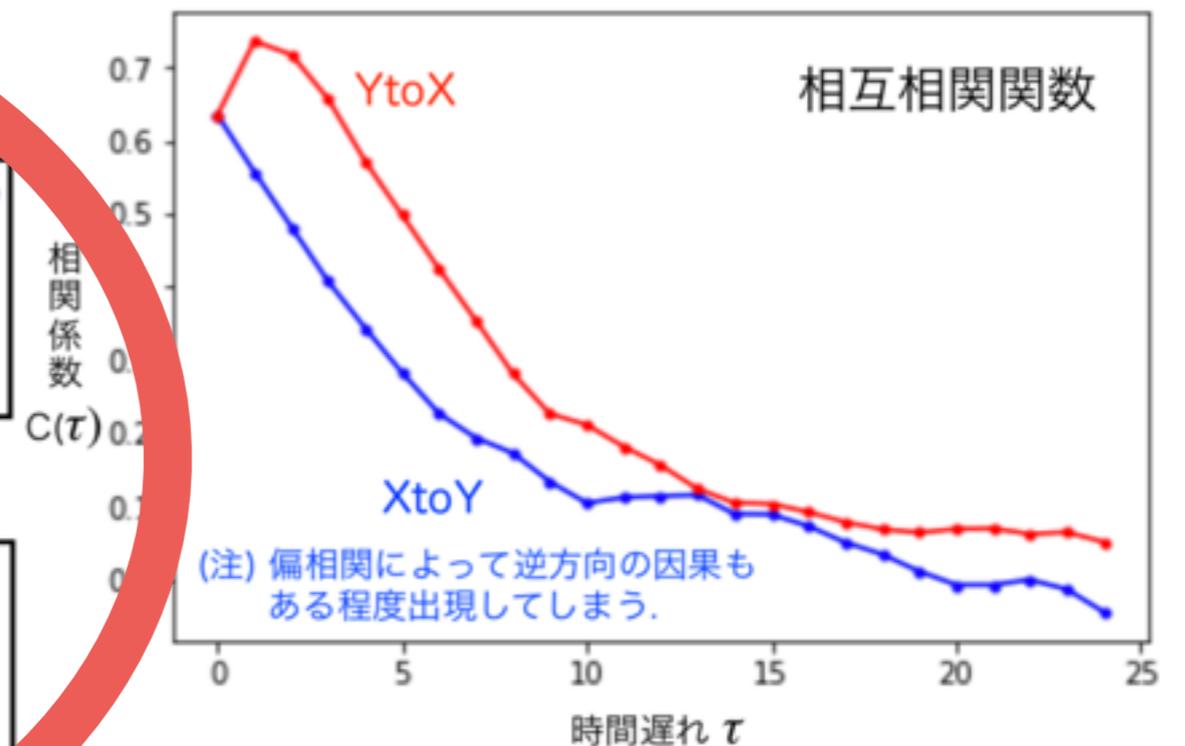
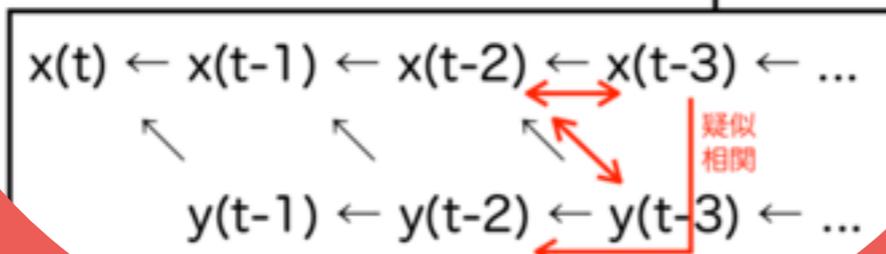
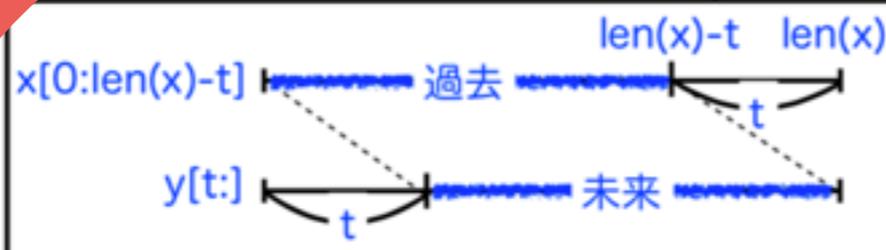
plt.plot(x, 'r')
```



```
XtoY = []
for t in range(0,25):
    tmp = np.corrcoef(x[0:len(x)-t], y[t:])[0,1] # yが未来
    XtoY.append(tmp)
```

```
YtoX = []
for t in range(0,25):
    tmp = np.corrcoef(y[0:len(y)-t], x[t:])[0,1] # xが未来
    YtoX.append(tmp)
```

```
plt.plot(XtoY, 'b.-')
plt.hold # 重ね表
plt.plot(YtoX, 'r.-')
```



[ポイント4] 追加情報です。