Mathematica(ver11)+SIR理論 による Covid-19 のデータの分析

§2. 拡大再生産数 R_t と γ を求める

テーマはあくまでも「*Mathematica*によるデータ分析の試み」です. *Mathematica* に対しても Covid-19 に関しても 専門家ではないので，間違いがあるかもしれません.　^_^;

2020年5月13日(5月19日改定).　　　生越 茂樹. oh.shigeki@gmail.com

```
ClearAll["Global`*"];
(*時系列リストdata={v1,v2,...vn}の変域tをbrに従って分割し,dataの区分近似直線関数を作る.
関数のリストを作成する関数がfuncs,傾きのリストを作成する関数がslopes.
  funcsを組み合わせdataの折れ線近似を作る関数が multifit.
  slopesを組み合わせdataの傾きの棒グラフを作る関数が multislope. 共にtの式*)
zip[A_List, B_List] := Partition[Riffle[A, B], 2]
slopes[data_, br_] := Module[{a, b}, Table[
   a /. FindFit[zip[Range[br[[k]], br[[k + 1]]], Take[data, {br[[k]], br[[k + 1]]}]], a x + b, {a, b}, x], {k, 1, Length[br] - 1}]]
pieces[data_, br_] := Table[Fit[zip[Range[br[[k]], br[[k + 1]]], Take[data, {br[[k]], br[[k + 1]]}]], {1, x}, x],
  {k, 1, Length[br] - 1}]
multifit[data_, br_] := Which @@ Flatten[Table[{br[[k]] ≤ x <= br[[k + 1]], pieces[data, br][[k]]}, {k, 1, Length[br] - 1}]] /.
  {x → t}
multislope[data_, br_] := Which @@ Flatten[Table[{br[[k]] ≤ x <= br[[k + 1]], slopes[data, br][[k]]}, {k, 1, Length[br] - 1}]] /.
  {x → t}
```

■ 0. SIR理論

s = Susceptible (感染可能人数), i = Infected (感染者), r = Recovered (回復者, 死者含む), c = Confirmed Case = i + r とすると (この r と R_t は全く異なる.また s, i, r, c をプログラム中では sus, inf, rec, cases とも書く), SIR理論では,

$$\frac{ds}{dt} = -\beta s * i,\quad \frac{di}{dt} = \beta s * i - \gamma i,\quad \frac{dr}{dt} = \gamma i$$

Nを全人口とすると i + r + s = c + s = N だから,

$$\frac{ds}{dt} = -\beta s * i,\quad \frac{dc}{dt} = \beta s * i,\quad \frac{di}{dt} = \beta s * i - \gamma i,\quad \frac{dr}{dt} = \gamma i \qquad \cdots (\#1)$$

故に $R_t = \frac{s\beta}{\gamma}$ と定義すると $\left(\text{これから } R_0 = \frac{N\beta}{\gamma},\ R_t = R_0 * \frac{s}{N} \text{ となる}\right)$,

$$\frac{ds}{d(\gamma t)} = -R_t * i,\quad \frac{dc}{d(\gamma t)} = R_t * i,\quad \frac{di}{d(\gamma t)} = R_t * i - i = (R_t - 1)\, i,\quad \frac{dr}{d(\gamma t)} = i \qquad \cdots (\#2)$$

さらに 時間の単位を1 / γ日に変えて γt = t’ とすると (例えばγ = 0.1 のときは 10 日ごとの変化を考えると),

$$\frac{ds}{dt'} = -R_t * i,\quad \frac{dc}{dt'} = R_t * i,\quad \frac{di}{dt'} = (R_t - 1)\, i,\quad \frac{dr}{dt'} = i$$

左の2式からR_t は 感染期間 (回復期間) 毎に, 1 人の感染者が2次感染させる人数 (即ち, 1 人の感染者が2次感染させる人数) となる事がわかる.
(#2 より R_t = dc / dr となることからも同じ事が分かる).
また右から2番めの式から $R_t > 1$ のとき I は増加, $R_t < 1$ のとき I は減少と分かる.

さらに $m_r = \frac{d(\log r)}{dt}$, $m_c = \frac{d\mathrm{Log}[c]}{dt}$ と定義すると,

$$m_r = \frac{d(\log r)}{dt} = \frac{1}{r} * \frac{dr}{dt} = \frac{\gamma * i}{r} \quad \therefore \gamma = \frac{r * m_r}{i},\qquad m_c = \frac{d\mathrm{Log}[c]}{dt} = \frac{1}{c} * \frac{dc}{dt} \quad \therefore \frac{dc}{dt} = c * m_c. \quad \cdots (\#3)$$

故に

$$R_t = \frac{s\beta}{\gamma}\left(= \frac{1}{i * \gamma}\frac{dc}{dt}\right) = \frac{c * m_c}{i * \gamma} = \frac{c * m_c}{r * m_r}\left(= \frac{dc}{dr}\right) \quad \cdots (\#4)$$

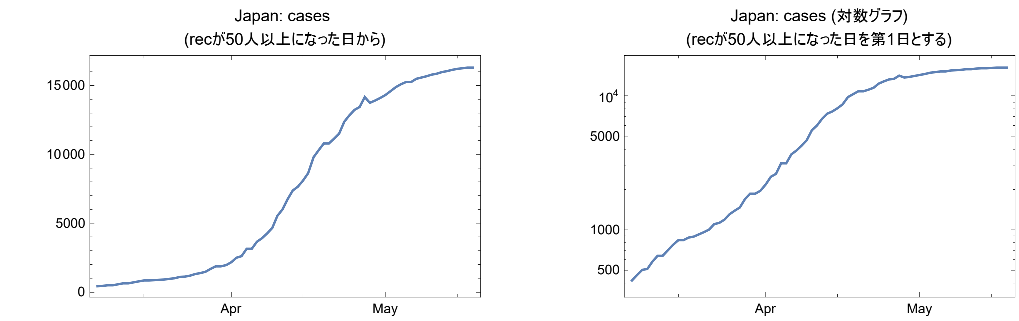
■ 1. 前章で Export したデータ(nation,breaks,case,rec,deaths)の Import

■ 2. ConfirmedCases の対数グラフの傾き m_c の値を求める.

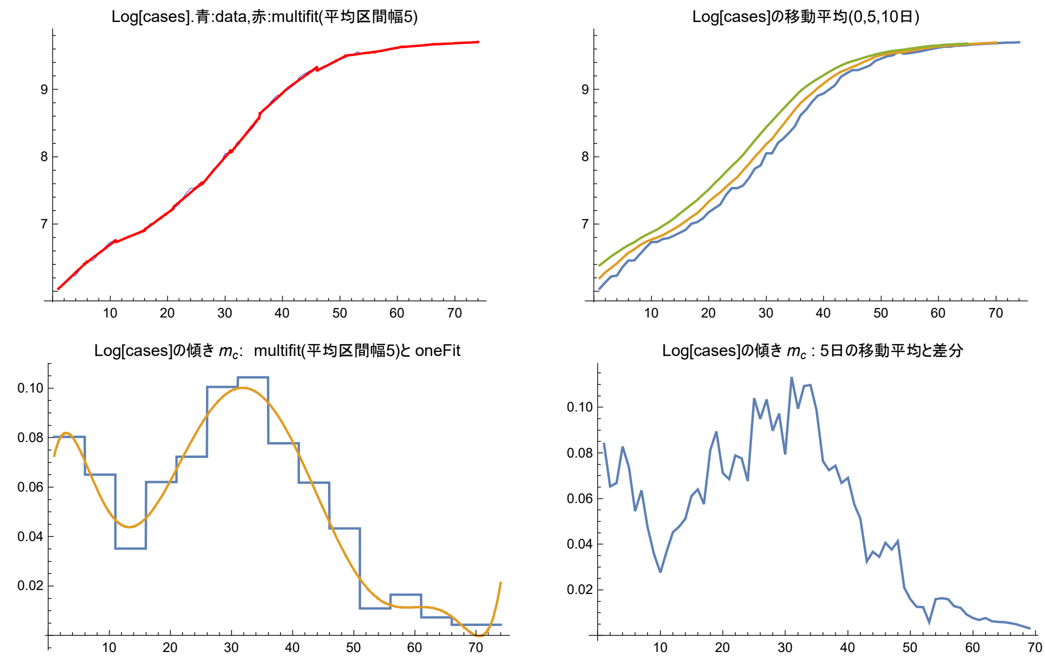
前章の繰り返しになるが, 「移動平均+差分を使う方法」と「全体のFit関数の微分(oneFit)」と「breaksに従って期間を分割してFitを使う方法(multifit)」を比較する.

In[160]:=
```
casesTL = TimeSeries[cases, {dates}] // Quiet;
GraphicsRow[{DateListPlot[casesTL, PlotLabel → nation <> ": cases \n(recが50人以上になった日から)", PlotRange → All],
  DateListLogPlot[casesTL,
   PlotLabel → nation <> ": cases (対数グラフ)\n(recが50人以上になった日を第1日とする)", PlotRange → All]}, ImageSize → 700]

logCmultifit[t_] = multifit[Log[cases], breaks];(*Log(Case)の折れ線近似*)
mc[t_] = multislope[Log[cases], breaks];(*mcの関数. multiFit*)
mr[t_] = multislope[Log[rec], breaks];(*mrの関数. multiFit*)
mcByOneFit = D[Fit[Log[cases], Table[t^k, {k, 0, 10}], t], t];(*全体にFitを使用し微分-onefit*)
mcByMovingAverage = Differences[MovingAverage[Log[cases], 5]];(*mcの5日の移動平均*)

GraphicsGrid[
 {{Show[{ListLinePlot[Log[cases], PlotStyle → {Thin, Blue}], Plot[logCmultifit[t], {t, 1, period}, PlotStyle → Red]},
    PlotLabel → "Log[cases]." <> "青:data,赤:multifit(平均区間幅" <> ToString[mhaba] <> ")"],
   ListLinePlot[{Log[cases], MovingAverage[Log[cases], 5], MovingAverage[Log[cases], 10]},
    PlotLabel → "Log[cases]の移動平均(0,5,10日)"]},
  {Plot[{Tooltip[mc[t], "multifit"], Tooltip[Evaluate[mcByOneFit], "onefit"]}, {t, 1, period},
    PlotLabel → "Log[cases]の傾き mc:" <> " multifit(平均区間幅" <> ToString[mhaba] <> ")と oneFit"],
   ListLinePlot[mcByMovingAverage, PlotLabel → "Log[cases]の傾き mc : 5日の移動平均と差分"]}}, ImageSize → 700]
```

Out[161]=



Out[167]=



3つのグラフの傾向は同じだが, 移動平均の方は,やや変動が大きく,かつ変域が左にずれる. また onefit は両端が大きく外にずれていく. $R_t=\frac{c*m_c}{i*\gamma}$ **を使うと** R_t **も同じく両端が外にずれることになる. onefitは表示には使えないが, 微分方程式に使うことは出来る. 両方に使うため multifit を使う. 即ち** `mc = mc[t], mr = mr[t]` **(前章) とする.**

■ 3. m_c と γ から 再生産数 R_t を求める. R_t<1 のときは収束，R_t>1 のときは拡大

(#3), (#4)より $\gamma = m_r * \frac{r}{i}$,　$R_t = \frac{S\beta}{\gamma} = \frac{c*m_c}{i*\gamma} = \frac{c*m_c}{r*m_r}$ $(=\frac{dc}{dr})$

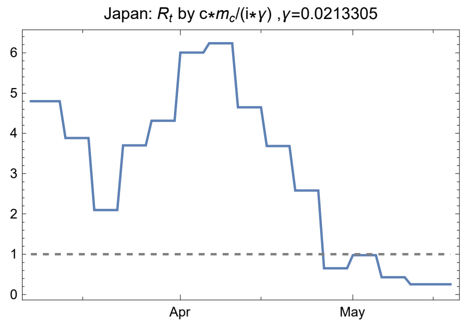
γが固定された時の R_t の式

m_c=mc[t] と決めたが，R_t の計算方法は $R_t = \frac{c*m_c}{i*\gamma}$, $R_t = \frac{c*m_c}{r*m_r}$ の2つある．数学的には同じだが，実際にはデータの変動により，2つの式の値は大きく違う．ここでは基本方針として，

「できるだけ変動が少なく，かつその R_t で計算した cases が　データのc と一致する」

ように選ぶ．まず変動の少ないのは c/i と γ を平均値で考え，$R_t = \frac{c*m_c}{i*\gamma}$ を使った式だろう.

In[168]:=
```
cOveri = TrimmedMean[N[cases / inf], {0.2, 0.2}];
γ = γ0 = TrimmedMean[rec / inf * Table[mr[t], {t, 1, period}], {0.2, 0.2}] // N;
Rt[t_] = mc[t] * cOveri / γ;
RtTL = TimeSeries[Table[Rt[t], {t, 1, period}], {dates}] // Quiet;
constTL = TimeSeries[1^Range[period], {dates}] // Quiet;

Show[{DateListPlot[{RtTL, constTL}, PlotStyle → {Line, {Gray, Dashed}},
   PlotLabel → nation <> ": Rt by c*mc/(i*γ) ,γ=" <> ToString[γ]]}, ImageSize → 300]
```

Out[173]=


嬉しいことに 最近では $R_t < 1$ となっているようだ．実際には γ の値はもっと大きいと思われる（前章）.
そして，$R_t = \frac{c*m_c}{i*\gamma}$ より，R_t の値はもっと下がるはず.
（しかしc を変えないでγ のみを大きくするとiは減るので反比例ではない．でもこの値より小さくなることは間違いない）.

4.微分方程式の解と比較することにより, R_t と γ を求める

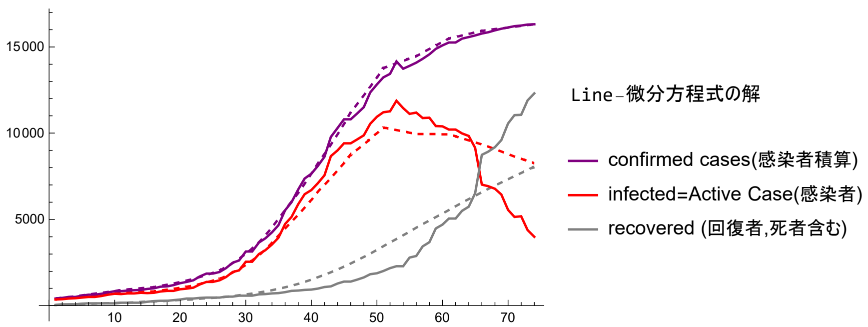
4-1. R_t=Rt[t] として良いか?

先のRt(=mc[t]*c/(i*γ)) は 「丸め」や「近似」を繰り返して得た式である. 正しい保証はない.
前の sub chapter で述べた基本方針に従って「この Rt を微分方程式(#2)に使ったとき
得られる結果式の c(t),i(t),r(t) がデータに合っているかどうか」を調べる. 以下 γ として
データから得られた平均値を利用する.

In[174]:=
```
cOveri = TrimmedMean[N[cases / inf], {0.2, 0.2}];
γ = γ0 = Round[TrimmedMean[rec / inf * Table[mr[t], {t, 1, period}], {0.2, 0.2}], 0.001];
Rt[t_] = mc[t] * cOveri / γ0; (*以上3つは直前のsub chapter と同じ*)
u = 0.98;
R0[t_] := u Rt[t];
cByData = Tooltip[cases, "cases by Data"];
dByData = Tooltip[deaths, "deathes by Data"];
iByData = Tooltip[inf, "inf by Data"];
rByData = Tooltip[rec, "rec by Data"];
```

In[183]:=
```
Module[{sol},
 sol = NDSolve[(*←微分方程式*)
   {i'[t] == R0[t] i[t] * γ0 - γ0 * i[t], r'[t] == γ0 * i[t],
    i[1] == inf[[1]], r[1] == rec[[1]]}, {i, r}, {t, 1, period}];
 Show[
  Plot[{Evaluate[i[t] /. sol] + Evaluate[r[t] /. sol],
    Evaluate[i[t] /. sol],
    Evaluate[r[t] /. sol]}, {t, 1, period}, PlotStyle → {{Purple, Dashed}, {Red, Dashed}, {Gray, Dashed}},
   PlotLegends → " Line-微分方程式の解 \n", PlotRange → All],
  ListLinePlot[{cByData, iByData, rByData},
   PlotStyle → {Purple, Red, Gray},
   PlotLegends → {"confirmed cases(感染者積算)", "infected=Active Case(感染者)", "recovered (回復者,死者含む)"}]]]
```

Out[183]=



上のプログラムで見るように, R_t = 0.98 * Rt[t] とすると, cについてはほぼ完全に一致する.
i, rについてはかなりずれているが, これは 計算されたγ の値が 本当はもっと大きいからだと
推定している. (前章と次のsub chapter参照)

従って 日本については R_t = Rt[t] として良い. そして実は世界の殆どの国に対し
「Rt[t] を使って解いた方程式の解」と「データ」は, c についてはほぼ一致する.
 (この章の最後のManipulateで確認できます.)

4-2. γ と R_t の正しい値を求める

(この節では 回復者は死亡者を除く. ただし recやr は 死亡者も含むとする)
上の点線と実線のグラフを比較すると r のグラフはやや不自然である. また
γ= γ_0 =0.021(5/19現在)というのは小さすぎる (回復期間=50日は長すぎる). かつ
回復者/死亡者=6.5なのに, 死亡率がたった2%というのもおかしい. (13%のはず?)
それは本当の回復者はもっと多いからではないか? (前章)

前章より「cases=C*a_t」とざっくり近似したとすると a=1.06

In[184]:=
```
a = Module[{m, n, t, sol}, sol = FindFit[Log[cases], m t + n, {m, n}, t]; Exp[m] /. sol];
Print[Style["概算で cases=c a^t, 但し a=" <> ToString[a], Blue, Larger]]
```

概算で cases=c a^t, 但し a=1.05807

平均の回復日数をp(=1/γ), 死亡者が死亡するまでの平均期間をq日,死亡率を2%とすると

a^{p-q}= $\frac{0.98}{0.02}$* 死者/回復者=49 * 死者/回復者

故に　p-q= $\log_a$ (49 * 死者/回復者)

∴　q=1/γ- $\log_a$ (49 * 死者/回復者)　…(#5)

回復者/死亡者=6.5 を代入して計算すると, p=48日, q=16日 (5/19現在) となり おかしい.
しかし回復者の率が大きくなるとp, qは小さくなる.

以上も考慮し ここでは *「Confirmed Cases(感染者積算)とdeaths(死亡者積算)の2つは正しいと仮定し,*
γ のみを動かして, p, qの値と rec,inf さらに Rt の推移を見てみたい.」

(#1),(#3) より　$\frac{dc}{dt}$= c*m_c= β s*i = R_t*γ*i

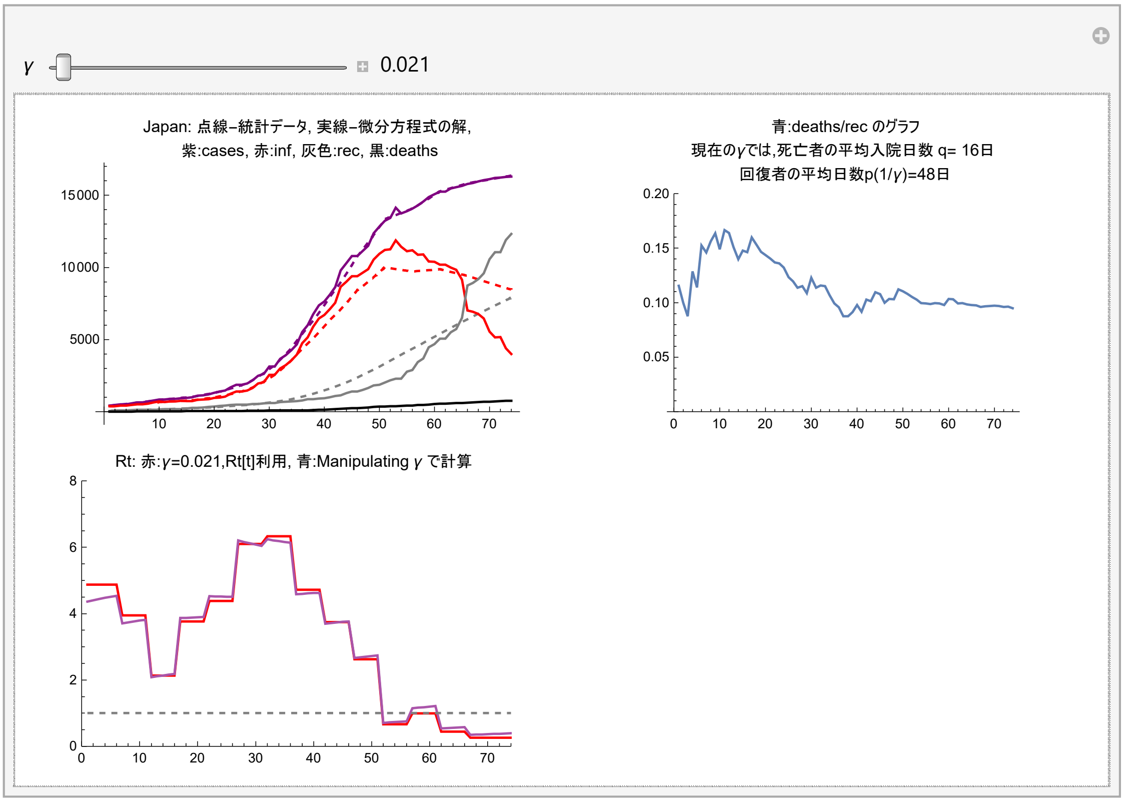
Confirmed cases は正しいと仮定するので

$\frac{dc}{dt}$= c*m_c で方程式を立て, その解の c[t],i[t] を使って　R_t= $c \star$mc[t]/ i/γ

とする. (先程の Rt=mc[t]*cOveri/γ と比較すると,共に, c/i を平均値に取るか, 時間tの関数と考えるかの違い)

In[186]:=
```
ClearAll[γ]

DynamicModule[{i, c, r, q, sol, rt1Table, rt2Table, cAssoc, cTable, γTable, dOverr},
 γTable = Table[γ, {γ, γ0, 0.1, 0.001}];
 cTable = Quiet[Table[
    sol = NDSolve[(*微分方程式*)
      {i'[t] == 0.985 (i[t] + r[t]) mc[t] - γ * i[t], r'[t] == γ * i[t],
       i[1] == inf⟦1⟧, r[1] == rec⟦1⟧}, {i, r}, {t, 1, period}];
    {Evaluate[i[t] /. sol] + Evaluate[r[t] /. sol],
     Evaluate[i[t] /. sol], Evaluate[r[t] /. sol]}, {γ, γ0, 0.1, 0.001}]];
 cAssoc = AssociationThread[γTable → cTable];
 Manipulate[
  c = cAssoc[γ]⟦1⟧; i = cAssoc[γ]⟦2⟧; r = cAssoc[γ]⟦3⟧;
  dOverr = Transpose[Table[deaths⟦t⟧ / r, {t, 1, period}]];
  q = Round[1 / γ - First[Log[a, 49 * TrimmedMean[Table[deaths⟦t⟧ / (r - deaths⟦t⟧), {t, 1, period}], {0.2, 0.2}]]]];
  rt1Table = Table[Rt[t], {t, 1, period}]; (*γ=γ0に対するRt*)
  rt2Table = Transpose[Table[c * mc[t] / i / γ, {t, 1, period}]] // First;
  (*Manipulating γ に対するRt*)
  GraphicsGrid[{
    {
     Show[Plot[Evaluate[cAssoc[γ]], {t, 1, period},
       PlotStyle → {{Purple, Dashed}, {Red, Dashed}, {Gray, Dashed}, {Black, Dashed}}, PlotRange → All],
      ListLinePlot[{cByData, iByData, rByData, dByData}, PlotStyle → {Purple, Red, Gray, Black}],
      PlotLabel → nation <> ": 点線-統計データ, 実線-微分方程式の解,\n 紫:cases, 赤:inf, 灰色:rec, 黒:deaths"],

     ListLinePlot[ dOverr, PlotRange → {0, 0.2},
      PlotLabel → " 青:deaths/rec のグラフ\n現在のγでは,死亡者の平均入院日数 q= " <> ToString[q]
        <> "日\n 回復者の平均日数p(1/γ)=" <> ToString[Round[1 / γ]] <> "日"]
    }, (*ここまでRow1*)
    {
     ListLinePlot[ {rt1Table, rt2Table, Table[1, {period}]},
      PlotRange → {{0, period}, {0, 8}}, PlotStyle → {Red, Lighter[Purple], {Gray, Dashed}},
      PlotLabel → "Rt: 赤:γ=" <> ToString[γ0] <> ",Rt[t]利用, 青:Manipulating γ で計算 "]
    }(*ここまでRow2*)
   }, ImageSize → 700],
  {γ, γ0, 0.1, 0.001, Appearance → "Labeled"},
  SaveDefinitions → True]]
```

Out[187]=



[結論] まず R_t= c*mc[t]/ i/γ...(*) と置いて微分方程式を解くと,
c/iを平均値で近似した Rt[t] の式を使った時より c[t] とcはさらに一致している. (左上図)

次に左下図を見ると $\gamma = \gamma 0$における2つのR_t のグラフはほぼ一致している.
即ち $R_t = c * \mathrm{mc}[t]/ i/\gamma$ において c/i は平均値を使っても時間tの式で計算しても大差ない!
（r1Tableとr2Tableの定義を参照. この結果は少し驚きであるが...)

次にγの妥当な値を決定する. これは「総合的判断」が必要になる. また現在日時にも依る.
5/19現在, 死亡者の平均入院日数を見ると γ＝0.027ぐらいが妥当だと思う(9日間).
しかしこれでは 5/19現在 「データのrec(統計rと書く)」>「解曲線のrec(計算rと書く)」となりおかしい.
(5/13現在では 入院日数からγ=0.033とすると,「統計rec<計算rec」だったので 丁度良かった.)
そして 5/19現在で「統計rec=計算rec」とすると γ=0.05前後となる.
しかしこの時 死亡者の平均入院日数は(-1)日となってしまう.

この理由は最近のcのカーブがなだらかに減少(aが減少)しているからだと思われる.
故にcのカーブが「ざっくり見ても指数関数」とは言えなくなる.
先の式は「ざっくり見て指数関数」というのに依存しているので, カーブが平らになると
式を改良しないと, この方法では適切なγを求めることはできないだろう.
(そんなに難しくないと思うので, 時間があればやってみたい.)

たった一週間でこんなに変わって驚きだが, 5/19現在では γ=0.05 としたい.
そうすると現時点における 計算r と 統計r が一致する. そして過去に「計算r > 統計r」だったのは
適切に統計されていなかったためと考えることが出来る.

以上から 5/19現在に置いて 妥当なγは γ=0.05 としたい. そして そのγ と方程式の解に基づいて計算した

$$R_t = c * \mathrm{mc}[t]/ i/\gamma$$

を採用したい. (c/i は時間平均でも, 瞬間値でもR_tの値は大差ない)

$\boldsymbol{R_t}$は 残念であるが, γ がほぼ2倍になっても 反比例して1/2倍にはならない.
$\boldsymbol{R_t}$>>1のときはかなり変化が大きいが, $\boldsymbol{R_t}$≤1のときは, あまり変化しない.

＊なお私が γを決めた時の Manipulate の実行画面を下に置いておきます.

Out[188]= 5月13日の実行画面

Out[189]= 5月19日の実行画面

5. R_t の表示関数の作成

いろいろ分析したので長くなりましたが，以上から 統計データを完全に信ずるなら
（γ = γ0 = 0.021（5 / 19 現在）で良いなら）R_t を求めるのは簡単です．
r の統計データを信じないのなら（γ = γ0は良くないと思うなら），
微分方程式を解かないと R_t は求まりません．でもMathematica では比較的簡単に求められます．
なお，使うには「zip, slopes, multislope」（notebook最上部）が必要です．

以下のプログラムは
（1） casesが正しいと仮定して，mc[t] を求める．
（2） cases を動かさずに γを与えて（データから計算されたγの値は使わないで）微分方程式を解いて r, i, c を求める．
（3） R_t = c * mc[t] / i / γ で R_t を求める．(c / i は平均値ではなく，時間tの関数）．

の手順で作成しています．最後の引数は ラベル（PlotLabel）です．必要ない時は入力不要です．
casesやrecやdates は整形前でも大丈夫です．また見やすいように最後に Fit で平滑化しています．
あまり複雑な関数を作ったことがないので使いづらいと思います．プログラムを見て適当に書き直して下さい．
（こんな作り方をしてたら，何種類も作らないといけないので，なにか高度な良い方法があるはず．）

入力が「γ , cases ,rec, dates (,Label)」のとき，R_t の表示関数の作成．

In[190]:=
```
showRt[gamma_, cases0_List, rec0_List, dates0_List, label_String: ""] := Module[
  {day1, dayl, dates, cases, rec, inf0, dayInfLess10, cc, ii, haba, period, breaks, mc, sol, rtTable, rtTL, constTL,
   rtTableFit},
  day1 = First[FirstPosition[ rec0 - 10, _?Positive]]; (*rec>10になってから解析*)
  inf0 = cases0 - rec0;
  dayInfLess10 = Flatten[Position[ inf0 - 10, _?Negative]];
  dayl = Min[Select[dayInfLess10, # > day1 &] - 1, Length[cases0]]; (*inf<10になると中止*)
  cases = Take[ cases0, {day1, dayl}];
  rec = Take[ rec0, {day1, dayl}];
  dates = Take[ dates0, {day1, dayl}];
  period = Length[cases]; haba = 5; (*期間分割幅*)
  breaks = Table[1 + haba k, {k, 0, (period - 1) / haba - 1}] // Append[period];
  mc[t_] = multislope[Log[cases], breaks];
  sol = NDSolve[ (*微分方程式*)
     {i'[t] == 0.99 (i[t] + r[t]) mc[t] - gamma * i[t], r'[t] == gamma * i[t],
      i[1] == Round[cases[[1]] / cOveri], r[1] == Round[cases[[1]] (1 - 1 / cOveri)]}, {i, r}, {t, 1, period}];
  cc = Evaluate[i[t] /. sol] + Evaluate[r[t] /. sol];
  ii = Evaluate[i[t] /. sol];
  rtTable = First[Transpose[Table[mc[t] cc / ii / gamma, {t, 1, period}]]];
  (*rtTL=TimeSeries[rtTable,{dates}]//Quiet;*) (*平滑化していない*)
  rtTableFit = Fit[rtTable, Table[t^k, {k, 1, Floor[period / 4]}], t] // Evaluate;
  rtTL = TimeSeries[Table[rtTableFit, {t, 1, period}], {dates}] // Quiet; (*平滑化*)
  constTL = TimeSeries[Table[1, {period}], {dates}] // Quiet;
  DateListPlot[{rtTL, constTL}, PlotStyle → {Line, Dashed}, Epilog → {Lighter[Purple], PointSize[Small],
     Tooltip[Point[{Last[dates], Last[rtTable]}], ToString[Round[Last[rtTable], 0.1]]], White,
     PointSize[Medium], Tooltip[Point[{Last[dates], 0}], ToString[Last[dates]]]}, PlotLabel → label]]
```

6. 東京,北海道,大阪と日本全体の R_t を図示

表示関数(showLocalRt)の作成

東京のデータは荻原さんがGitHubに公開されておられる素データをDownload します.
(https://github.com/kaz-ogiwara/covid19/blob/master/data/prefectures.csv)
こちらは3月11日からのデータになります.
先程の showRt を使って γと地方名を入力すると, R_t が描けるような関数 `showLocalRt` を作ります.

In[191]:=
```
japanData = Import["https://raw.githubusercontent.com/kaz-ogiwara/covid19/master/data/prefectures.csv"];
japanDates = Select[japanData, Part[#, 5] == "Tokyo" &][[All, 1 ;; 3]];
```

In[193]:=
```
showLocalRt[gamma_, region_String] := Module[
  {localData, localCases, localDischarged, localDeaths, localRec, localInf},
  localData = Select[japanData, Part[#, 5] == region &];
  localCases = localData[[All, 6]] /. {"" → 0};
  localDischarged = localData[[All, 8]] /. {"" → 0};
  localDeaths = localData[[All, 9]] /. {"" → 0};
  localRec = localDischarged + localDeaths;
  localInf = localCases - localRec;
  showRt[gamma, localCases, localRec, japanDates, "Rt:" <> region <> " : γinput=" <> ToString[gamma]]]
```

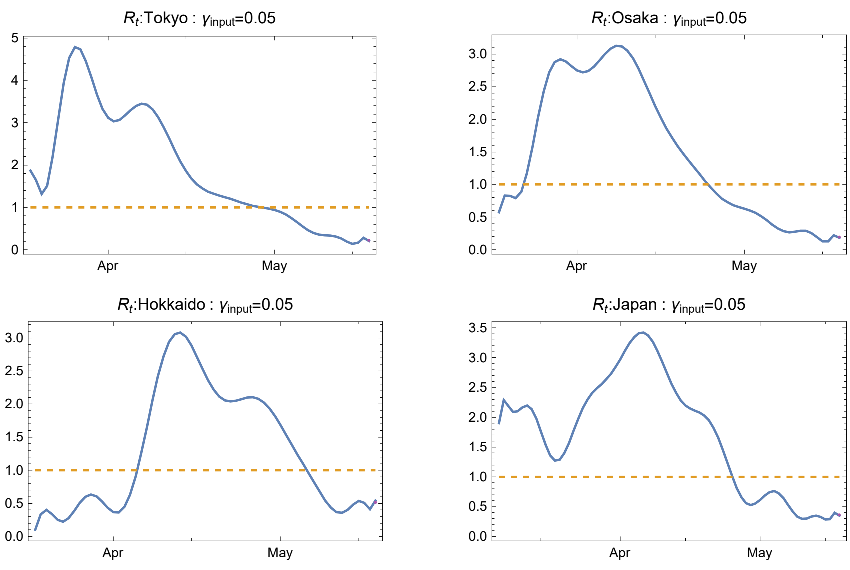
東京,北海道,大阪と日本全体のR_t の比較

γ = 0.05 で計算します. また最近の R の値を Tooltip しました.

In[194]:=
```
γ1 = 0.05

GraphicsGrid[{{showLocalRt[γ1, "Tokyo"], showLocalRt[γ1, "Osaka"]},
  {showLocalRt[γ1, "Hokkaido"], showRt[γ1, cases, rec, dates, "Rt:" <> nation <> " : γinput=" <> ToString[γ1]]}}]
```

Out[194]= 0.05

Out[195]=



3月11日からなので,北海道の2月のピークは見えません. 傾向は意外なほど違いますが,
どちらも最近は余裕で R < 1 となっているようです.

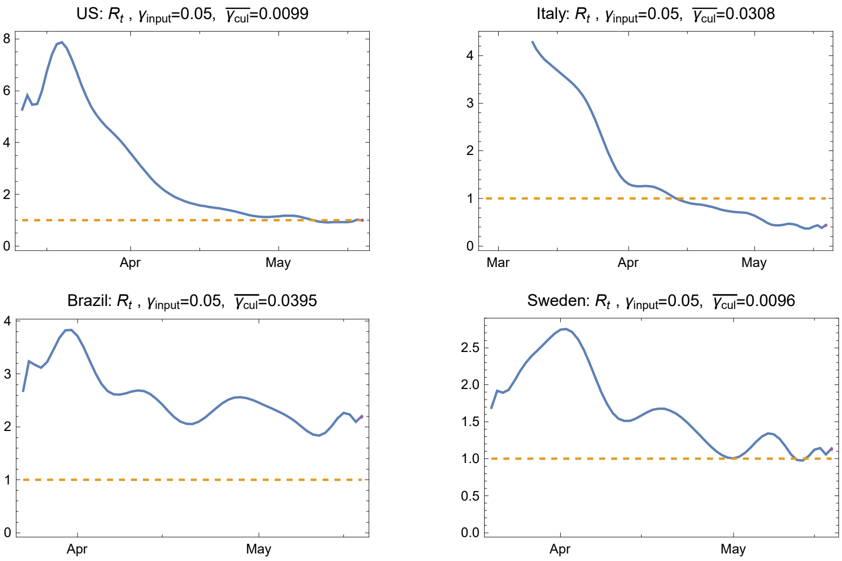
7. 世界の R_t を表示 (with Manipulate)

R_t の表示関数:showGlobalRt を作成（入力は γ と 国名 ）

US, Italy, Brazil, Sweden の R_t を見る

In[216]:=
```
γ1 = 0.05;
GraphicsGrid[{{
   showGlobalRt[γ1, "US"],
   showGlobalRt[γ1, "Italy"]},
  {showGlobalRt[γ1, "Brazil"],
   showGlobalRt[γ1, "Sweden"]}}, ImageSize → 600]
```

Out[217]=



USはいま一歩．Italyは 余裕で1 より小さいです． 逆にBrazilは「2.2前後」，
Swedenは 徐々に下がってきているもののまだ1より大きいです．

ただこれらは γ=0.05 という前提です．γの値は世界中でそんなに変わらないはずですが，
多少は医療設備や統計のとり方により変わると思います．
データから単純に計算したγの値がγ_{cul} ですが，国によって本当にまちまちです．
やはり私は 統計データのrecはあまり信頼できないと思います．
同じであるべき γ が国によってこんなに違うからです．

Italyを見ると R_t の値が最近30日間でも変わっているのがお分かりと思います．(SpainやGermanyなども同じ)
SIR理論に依ると β,γ は時間に依らないので 初期の段階では(s/人口=1 に近い間は) R_t は定数のはずです．
もちろん それは何も対策を講じなかった時の話で実際には各国で対策をするのでそれによって R_t は変わります．
ブラジルは何も対策してないようなので最近50日間は R_t は一定です．ブラジルはSIR理論に忠実です．
しかしイタリアは3月初旬からロックダウンをしているので，2~3週間は徐々に R_t が直線的に下がるのは理論通りです．
しかしその後も緩やかに下がり続けているのは何故でしょう？
スウェーデンは何も対策してないはずなのに何故下がっているのでしょう？
素人考えですが，これは SIR理論だけでは説明できないと思います．

R_t (& epidemic curve) の Manipulate