

Covid-19: elaborazione

Lorenzo Roi (18/05/2020)

Codice

Le istruzioni con sfondo colorato vanno aggiornate.

```
In[*]:= formatoISO = "20200518";
```

Preleva i file distribuiti dalla Protezione Civile e li salva nella cartella corrente. Gli stessi vengono rinominati mantenendo la loro estensione ma inserendo la data del giorno.

```
In[*]:= (*URLDownload["https://raw.githubusercontent.com/pcm-dpc/COVID-19/master/dati-json/dpc-covid19-ita-regioni.json", "datiRegionali" <> formatoISO <> ".json"];
URLDownload["https://raw.githubusercontent.com/pcm-dpc/COVID-19/master/dati-andamento-nazionale/dpc-covid19-ita-andamento-nazionale.csv", "datiNazionali" <> formatoISO <> ".csv"];*)
```

```
In[*]:= (* comando per selezionare la regione *)
selezionaRegione[datiNazionali_, regione_] := Select[datiNazionali, ({"denominazione_regione" /. #[[4]]} === regione) &];
(* funzione utilizzata dalle due successive per estrarre i dati di interesse a partire dai loro nomi *)
posizioneRecord[listaRecord_List] :=
  Flatten@SubsetPosition[{"data", "stato", "codice_regione", "denominazione_regione", "lat", "long", "ricoverati_con_sintomi", "terapia_intensiva", "totale_ospedalizzati", "isolamento_domiciliare",
    "totale_positivi", "variazione_totale_positivi", "nuovi_positivi", "dimessi_guariti", "deceduti", "totale_casi", "tamponi"}, listaRecord];
(* estrai i dati elencati in lista dall'insiemeDati della regione o della nazione *)
estraeDati[lista_List, insiemeDati_] := lista /. #[[posizioneRecord[lista]]] & /@ insiemeDati;
estraeDati[stringa_String, insiemeDati_] := Flatten[stringa /. #[[posizioneRecord[{stringa}]]] & /@ insiemeDati];
(* per disporre, eventualmente di un sottoinsieme dei dati regionali *)
estraeDatiRegionali[regione_String, lista_List, datiNazionali_] := estraeDati[lista, selezionaRegione[datiNazionali, regione]];
```

Importa i dati regionali e seleziona quelli della regione scelta (Veneto)

Aggiornare la regione e il numero di giorni dal 24: questi ultimi determinano l'estremo superiore dell'intervallo che viene rappresentato nelle varie figure. Eventualmente e, a seconda della regione, aggiornare i valori in ordinata di PlotRange.

```
In[*]:= giorniDal24 = 90;
regione = "Veneto";
```

```
In[*]:= datiRegionali = Import["datiRegionali" <> formatoISO <> ".json"];
datiRegione = selezionaRegione[datiRegionali, regione];
dataUltimoAggiornamento = StringTake[formatoISO, -2] <> "/" <> StringTake[formatoISO, {5, 6}] <> "/" <> StringTake[formatoISO, 4];
dataDal24 = DatePlus[DateObject[{2020, 2, 24}], giorniDal24]
```

```
Out[*]:=  Day: Sun 24 May 2020
```

```
In[ ]:= (* utile per definire le griglie verticali nei grafici*)
griglia = Table[i, {i, 0, giorniDal24, 2}];
(* definisce il colore di sfondo al primo grafico *)
celeste = RGBColor[{0, 0.518, 0.682}];
(* colore utilizzato nella tabella *)
celesteTenue = RGBColor[{0.913725, 1, 1}];
(* e il verde nelle figure 20 e 21 *)
verde = RGBColor[{0.113725, 0.760784, 0.270588}];
(* grafico standard dei dati e posizionamento etichetta*)
plotLineareDati[posizioneRelativa_List] := ListPlot[totaleCasi, PlotStyle -> {celeste, PointSize[0.012]}, PlotLegends -> Placed[{"N[t] totale positivi"}, posizioneRelativa]];
```

1. I dati della regione Veneto

```
In[ ]:= totaleCasi = estraeDati["totale_casi", datiRegione];

In[ ]:= fig01 = DateListPlot[totaleCasi, {2020, 2, 24}, PlotRange -> {Automatic, {-500, 20000}}, PlotStyle -> celeste, PlotLabel -> Style["Veneto: totale giornaliero casi positivi", Bold, FontSize -> 15],
  Joined -> False, Epilog -> Inset[Style["Aggiornato " <> dataUltimoAggiornamento, FontSize -> 13], Scaled@{.25, .75}], FrameLabel -> {"giorno", "totale positivi (cumulativo)"},
  AspectRatio -> 1 / 3, ImageSize -> {800, 267}, Filling -> Axis, FillingStyle -> celeste, Prolog -> {Lighter[LightBlue, .8], Rectangle[Scaled[{0, 0}], Scaled[{1, 1}]]}]
```

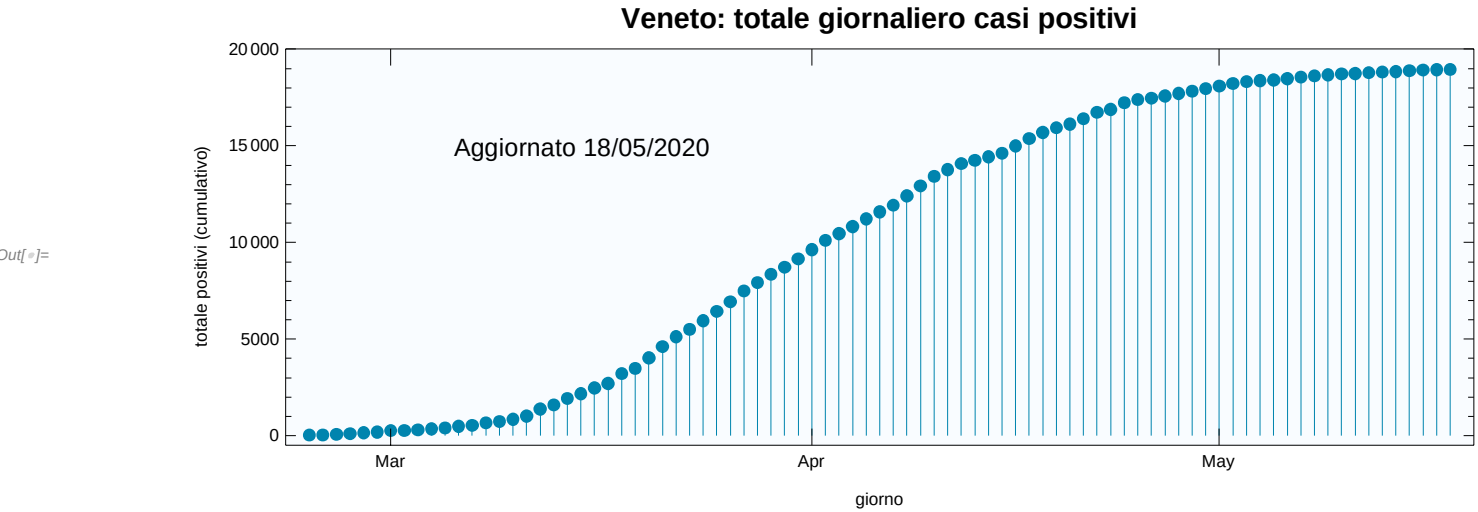


Figura 1

2. Funzione lineare

Genera la tabella di tutti i dati

```
In[*]:= coppieDati = Transpose[{Range[Length[totaleCasi]], totaleCasi}];
padDestra = PadRight[coppieDati, 100, {, }];
ok = (Flatten@padDestra[{{#, # + 20, # + 40, # + 60, # + 80}}]) & /@ Range[20];
Text@Grid[Prepend[ok, {" t (day)", "N[t]", "t (day)", "N[t]", "t (day)", "N[t]", "t (day)", "N[t]", "t (day)", "N[t]"}],
  Alignment -> {Right}, ItemStyle -> {Automatic, Automatic, {{1, 1}, {1, 10}} -> Directive[FontFamily -> "Helvetica", Bold, FontSize -> 13]}},
  Background -> {{2 -> celesteTenue, 4 -> celesteTenue, 6 -> celesteTenue, 8 -> celesteTenue, 10 -> celesteTenue}, None, {{1, 1}, {1, 10}} -> LightOrange},
  Frame -> All, Spacings -> {2, 1}, FrameStyle -> Directive[Dotted, Thickness[0.00001]]]
```

Out[*]=

t (day)	N[t]	t (day)	N[t]	t (day)	N[t]	t (day)	N[t]	t (day)	N[t]
1	33	21	2172	41	10824	61	17229	81	18845
2	43	22	2473	42	11226	62	17391	82	18889
3	71	23	2704	43	11588	63	17471	83	18928
4	111	24	3214	44	11925	64	17579	84	18941
5	151	25	3484	45	12410	65	17708	85	18950
6	191	26	4031	46	12933	66	17825		
7	263	27	4617	47	13421	67	17960		
8	273	28	5122	48	13768	68	18098		
9	307	29	5505	49	14077	69	18224		
10	360	30	5948	50	14251	70	18318		
11	407	31	6442	51	14432	71	18373		
12	488	32	6935	52	14624	72	18402		
13	543	33	7497	53	14990	73	18479		
14	670	34	7930	54	15374	74	18553		
15	744	35	8358	55	15692	75	18618		
16	856	36	8724	56	15935	76	18671		
17	1023	37	9155	57	16127	77	18722		
18	1384	38	9625	58	16404	78	18741		
19	1595	39	10111	59	16738	79	18782		
20	1937	40	10464	60	16881	80	18813		

Tabella 1

```
In[*]:= graf3Rette = Plot[{x + 2, 2 x + 3, 3 x + 4}, {x, 0, 30}, PlotStyle -> {Red, LightBlue, Green}];
graf3Succ = ListPlot[{Table[{i, i + 2}, {i, 1, 30}], Table[{i, 2 i + 3}, {i, 1, 30}], Table[{i, 3 i + 4}, {i, 1, 30}]},
  PlotLegends -> Placed[{"a[t] = 2 + (t-1) -> a(t) = 2 + t", "b[t] = 3 + 2(t-1) -> b(t) = 3 + 2t", "c[t] = 4 + 3(t-1) -> c(t) = 4 + 3t"}, {.3, .77}], PlotStyle -> PointSize[0.015]];
fig02 = Show[graf3Rette, graf3Succ, Frame -> True, GridLines -> Automatic, FrameLabel -> {"indice t", "valori della successione"}, PlotLabel -> Style["Tre successioni aritmetiche\ne corrispondenti rette", Bold]]
```

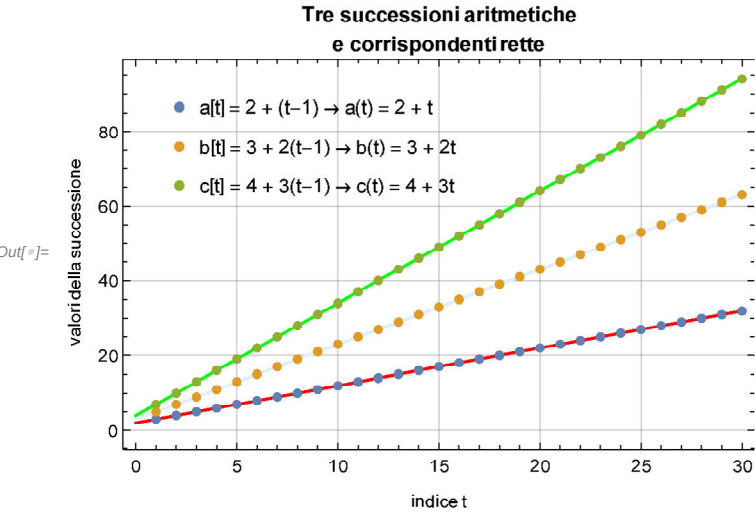


Figura 2
Minimi quadrati per il modello lineare ed equazione retta

```
In[*]:= mLineare = LinearModelFit[totaleCasi, t, t];
Normal[mLineare]

Out[*]:= -1950.64 + 285.607 t

In[*]:= grafRetta = Plot[Normal[mLineare], {t, 0, giorniDal24}, PlotStyle -> {Orange, Thickness[0.01]}, PlotLegends -> Placed[{"funzione lineare: f(t) = a t + b"}, {.67, .17}]];
(* *)
fig03 = Show[grafRetta, plotLineareDati[{.585, .28}], PlotRange -> {{0, giorniDal24}, {-2000, 21000}},
  PlotLabel -> Style["Veneto: totale dei casi positivi\n e modello lineare", Bold], Epilog -> {Inset["Aggiornato " <> dataUltimoAggiornamento, Scaled@{.25, .8}]},
  FrameLabel -> {"giorni dal 24 febbraio 2020", "N[t] = totale positivi (cumulativo)"}, GridLines -> {griglia, Automatic}, Frame -> True]
```

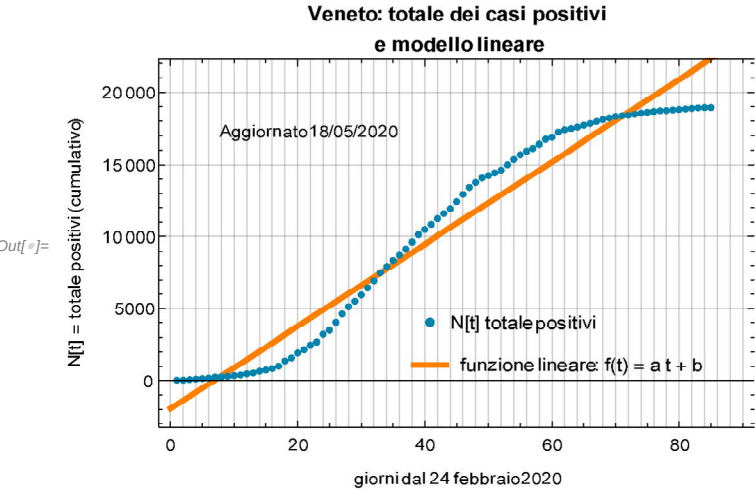


Figura 3

```

In[ ]:= casiCalcolati = Table[Normal[mLineare], {t, 1, Length[totaleCasi]}];
residui = totaleCasi - casiCalcolati;
fig04 = ListPlot[residui, PlotLabel -> Style["Residui (modello lineare): N[t]-n(t)", Bold], Filling -> {1 -> {Axis, {celeste, Lighter[Red, .5]}}},
  GridLines -> {griglia, Table[i, {i, -3000, 2500, 500}]}, Frame -> True, FrameLabel -> {"giorni dal 24 febbraio 2020", "residui"},
  PlotRange -> {{0, giorniDal24}, All}, PlotStyle -> {celeste, PointSize[0.012]}, Epilog -> {Inset["Aggiornato " <> dataUltimoAggiornamento, Scaled@{.65, .17}]}]

```

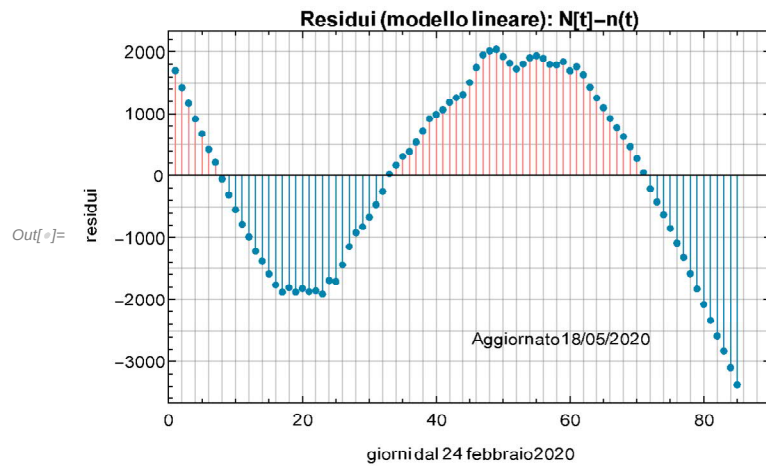


Figura 4

3. Modello esponenziale

```

In[ ]:= fig05a = TreePlot[{1 -> 2, 1 -> 3, 1 -> 4, 3 -> 5, 2 -> 6, 2 -> 7, 3 -> 8, 4 -> 9, 6 -> 10, 6 -> 11, 7 -> 12, 7 -> 13, 5 -> 14, 8 -> 15, 8 -> 16}, PlotTheme -> "IndexLabeled",
  VertexStyle -> {2 -> LightRed, 3 -> LightRed, 7 -> LightRed, 8 -> LightRed, 11 -> LightRed, 15 -> LightRed}, Frame -> True, ImageSize -> {350, 200}, FrameLabel -> Style["(a): N[t+1] = N[t] + 2", FontSize -> 14]];
fig05b = TreePlot[{1 -> 2, 1 -> 3, 2 -> 4, 2 -> 5, 3 -> 6, 3 -> 7, 4 -> 8, 4 -> 9, 5 -> 10, 5 -> 11, 6 -> 12, 6 -> 13, 7 -> 14, 7 -> 15},
  VertexStyle -> {2 -> LightRed, 4 -> LightRed, 6 -> LightRed, 8 -> LightRed, 11 -> LightRed, 15 -> LightRed, 12 -> LightRed},
  PlotTheme -> "IndexLabeled", Frame -> True, ImageSize -> {350, 200}, FrameLabel -> Style["(b): N[t+1] = 2*N[t]", FontSize -> 14]];
fig05 =
  GraphicsRow[
    {fig05a,
     fig05b}]

```

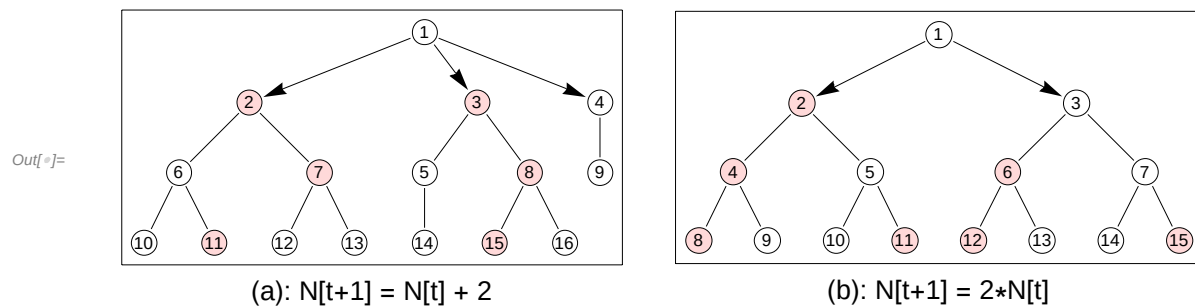


Figura 5

```
In[*]:= (* definizione ricorsiva dell'esempio *)
a[1] = 1; a[t_Integer /; t > 1] := 2 a[t - 1];
grafExp = Plot[2^(x - 1), {x, 0, 10}, PlotStyle -> Red, PlotLegends -> Placed[{"funzione esponenziale: f(t) = 2^{t-1}"}, {.35, .82}]];
grafProgr = ListPlot[Table[{i, a[i]}, {i, 1, 10}], PlotLegends -> Placed[{"progressione geometrica: a[t] = 2 a[t-1]"}, {.42, .7}], PlotStyle -> {PointSize[0.02], celeste}];
fig06 = Show[grafExp, grafProgr, Frame -> True, GridLines -> Automatic,
  FrameLabel -> {"indice t", "valori della successione"}, PlotLabel -> Style["Progressione geometrica e\n corrispondente grafico funzione esponenziale", Bold]]
```

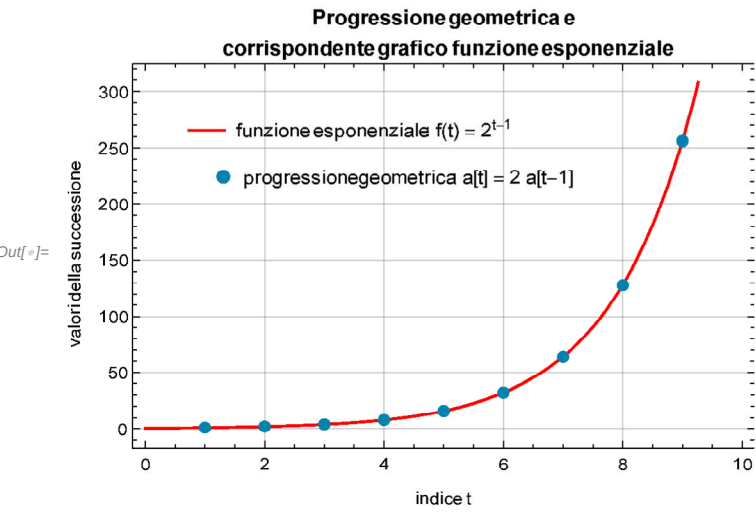


Figura 6
Minimi quadrati del modello esponenziale

```
In[*]:= esponenziale[t_] := a Exp[b {t - c}];
nlmEsponenziale = NonlinearModelFit[totaleCasi, esponenziale[t], {a, {b, 0.02}, c}, t]
Normal[nlmEsponenziale]
```

```
Out[*]:= FittedModel[ 33.328 e^{0.0231984 (199.024+t)} ]
```

```
Out[*]:= 33.328 e^{0.0231984 (199.024+t)}
```



```

In[ ]:= (*solo funzione esponenziale continua*)
esponenzialePlotLineare = Plot[Normal[nlmEsponenziale], {t, 0, giorniDal24},
  PlotStyle -> {Orange, Thickness[0.01]}, PlotLegends -> Placed[{"funzione esponenziale"}, {.24, .85}], GridLines -> {Table[i, {i, 2, giorniDal24, 2}], Automatic}};
(* *)
fig07 = Show[esponenzialePlotLineare, plotLineareDati[ {.21, .73}], PlotRange -> {{0, giorniDal24}, {0, 25000}},
  PlotLabel -> Style["Veneto: totale giornaliero casi positivi\n e modello esponenziale", Bold], Epilog -> {Inset["Aggiornato " <> dataUltimoAggiornamento, Scaled@{.75, .2}]},
  Frame -> True, FrameLabel -> {"giorni dal 24 febbraio 2020", "totale positivi (cumulativo)"}, GridLines -> {griglia, Table[i, {i, 0, 30000, 2000}]}}

```

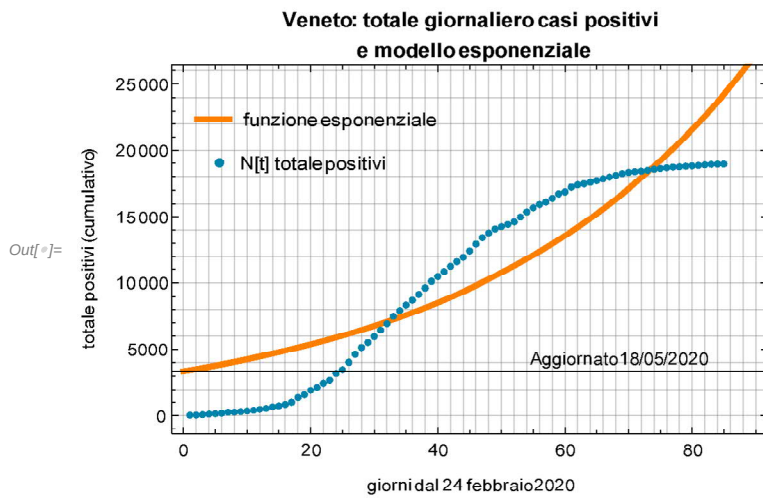


Figura 7

```

In[ ]:= residui = totaleCasi - Table[Normal[nlmEsponenziale], {t, 1, Length[totaleCasi]}];
fig08 = ListPlot[residui, PlotLabel -> Style["Residui (modello esponenziale): N[t]-n(t)", Bold], Filling -> {1 -> {Axis, {celeste, Lighter[Red, .5]}},
  GridLines -> {griglia, Table[i, {i, Round[Min[residui] - 500], Round[Max[residui] + 500], 500}]}, Frame -> True, FrameLabel -> {"giorni dal 24 febbraio 2020", "residui"},
  PlotRange -> {{0, giorniDal24}, All}, PlotStyle -> {celeste, PointSize[0.012]}, Epilog -> {Inset["Aggiornato " <> dataUltimoAggiornamento, Scaled@{.65, .17}]}}

```

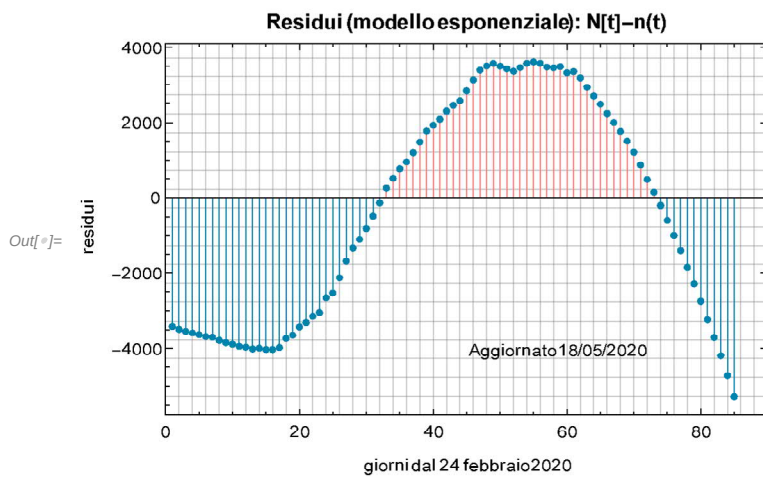


Figura 8

Modello esponenziale per i primi 22 giorni

```

In[ ]:= nlmEsponenziale22 = NonlinearModelFit[Take[totaleCasi, 22], esponenziale[t], {a, b, c}, t, Method -> "Newton"];
Normal[nlmEsponenziale22]

```

Out[]:= $2.65372 e^{0.167311 (19.0577 + t)}$

```
In[*]:= esponenzialePlotLineare22 = Plot[Normal[nlmEsponenziale22], {t, 0, giorniDal24},
  PlotStyle -> {Orange, Thickness[0.01]}, PlotLegends -> Placed[{"funzione esponenziale"}, {.3, .85}], GridLines -> {Table[i, {i, 2, giorniDal24, 2}], Automatic}};
(* *)
fig09 = Show[esponenzialePlotLineare22, plotLineareDati[ {.27, .7}], PlotRange -> {{0, 23}, {0, 3400}},
  PlotLabel -> Style["Veneto: totale giornaliero casi positivi\n e modello esponenziale (primi 22 giorni)", Bold], Frame -> True,
  FrameLabel -> {"giorni dal 24 febbraio 2020", "N[t] totale positivi (cumulativo)"}, GridLines -> {Table[i, {i, 0, 60, 1}], Table[i, {i, 0, 3400, 200}]}}
```

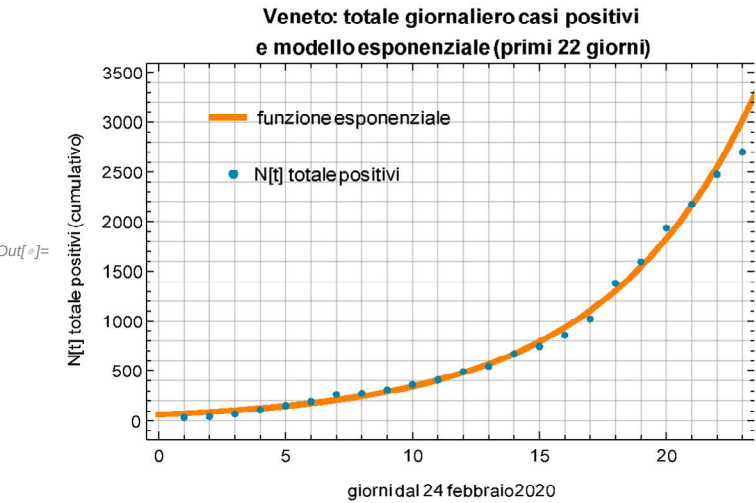


Figura 9

```
In[*]:= casiCalcolatiExp22 = Table[Normal[nlmEsponenziale22], {t, 1, 30}];
residui22 = Take[totaleCasi, 30] - Take[casiCalcolatiExp22, 30];
fig10 = ListPlot[residui22, PlotLabel -> Style["Residui (mod. esponenziale): N[t]-n(t)\n(primi 22 giorni)", Bold],
  Filling -> {1 -> {Axis, {celeste, Lighter[Red, .5]}}, GridLines -> {Table[i, {i, 0, 30, 1}], Table[i, {i, -150, 150, 25}]}, Frame -> True,
  FrameLabel -> {"giorni dal 24 febbraio 2020", "residui"}, PlotRange -> {{0, 23}, {-150, 150}}, PlotStyle -> {celeste, PointSize[0.015]}}
```

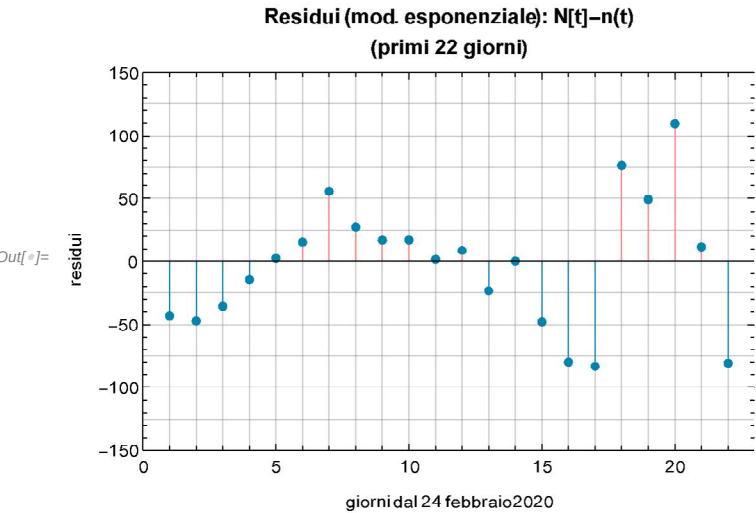


Figura 10


```

In[ ]:= fig11 = Show[esponenzialePlotLineare22, plotLineareDati[ {.27, .7}], PlotRange -> {{0, 44}, {0, 15000}}, PlotLabel -> Style["Veneto: totale giornaliero casi positivi\n e modello esponenziale", Bold],
  Frame -> True, FrameLabel -> {"giorni dal 24 febbraio 2020", "N[t] totale positivi (cumulativo)"}, GridLines -> {Table[i, {i, 0, 60, 1}], Table[i, {i, 0, 15000, 1000}]}]

```

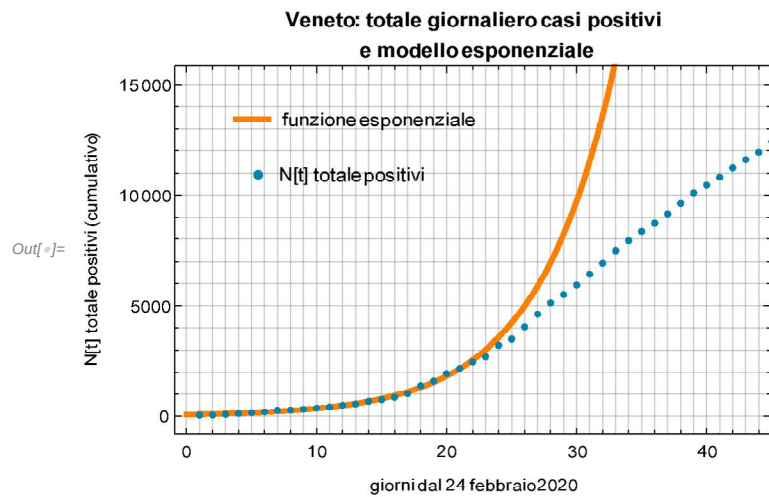


Figura 11

4. Modello logistico

```

In[ ]:= (* funzione esempio *)
expIniziale = Plot[1.5^(t - 1), {t, 1, 12.55}, PlotStyle -> {Orange, Thickness[0.01]}, PlotLegends -> Placed[{"curva esponenziale iniziale"}, {.7, .1}]];

In[ ]:= (* definizione ricorsiva dell'esempio: segue la tabella dei dati *)
succLogistica[1] = 1; succLogistica[n_Integer /; n > 1] := 1.5 succLogistica[n - 1] - .5 / 100 succLogistica[n - 1]^2;
(* datiEsemioLogistica si è ottenuta con datiEsemioLogistica = Table[{i, succLogistica[i]}, {i, 1, 22}] per evitare il ricalcolo ad ogni apertura del notebook *)
datiEsemioLogistica = {{1, 1}, {2, 1.495`}, {3, 2.2313248750000003`}, {4, 3.3220932590110315`}, {5, 4.927958370408715`}, {6, 7.2705136871106655`}, {7, 10.641468684293681`}, {8, 15.395998747646505`},
  {9, 21.908814234282097`}, {10, 30.46324064566174`}, {11, 41.05481581531512`}, {12, 53.15473421482544`}, {13, 65.60497247499448`}, {14, 76.8873966452678`}, {15, 85.772736153468`},
  {16, 91.87429289393981`}, {17, 95.60701086710253`}, {18, 97.7070136659425`}, {19, 98.82721790133039`}, {20, 99.40673186141038`}, {21, 99.70160609528386`}, {22, 99.85035785303006`}};

In[ ]:= plotEsemioLogistica = ListPlot[datiEsemioLogistica, PlotStyle -> {celeste, PointSize[0.015]}, PlotLegends -> Placed[{"valori della successione"}, {.69, .2}]];

(* *)
fig12 = Show[expIniziale, Graphics[{Purple, Thick, Dashed, Line[{1, 100}, {22, 100}]}], plotEsemioLogistica, Frame -> True, FrameLabel -> {"indice t", "popolazione N[t]"},
  GridLines -> {Range[22], Automatic}, PlotRange -> {{1, 22}, Automatic}, PlotLabel -> Style["Esemio di successione logistica\n e curva esponenziale iniziale", Bold]]

```

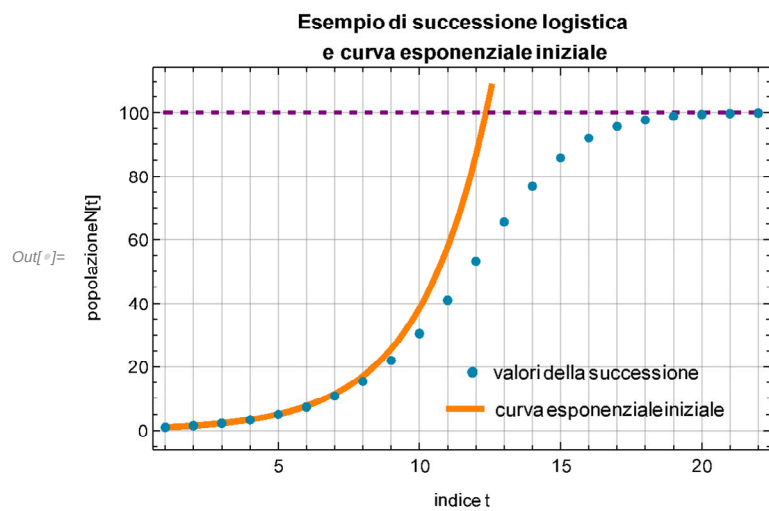


Figura 12

```
In[*]:= successioneLogPlot = ListLogPlot[datiEsempioLogistica, PlotStyle -> {celeste, PointSize[0.015]}, PlotLegends -> Placed[{"valori della successione"}, {.7, .4}]];
funzioneLogPlot = LogPlot[1.5^t-1, {t, 1, 22}, PlotStyle -> {Orange, Thickness[0.01]}, PlotLegends -> Placed[{"curva esponenziale iniziale"}, {.71, .3}]];
fig13 = Show[funzioneLogPlot, successioneLogPlot, PlotRange -> {{0, 22}, {0, Log[100]}}, Frame -> True,
  PlotLabel -> Style["Successione logistica\n e andamento iniziale esponenziale (logPlot)", Bold], FrameLabel -> {"indice t", "popolazione N[t]"}, GridLines -> {Range[22], Automatic}]
```

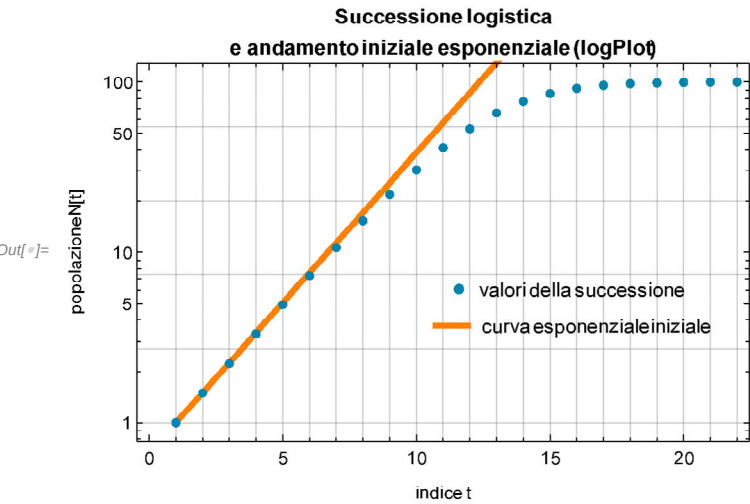


Figura 13

Minimi quadrati per la funzione logistica

```
In[*]:= logistica[t_] := M / (1 + β Exp[-α t]);
nlmLogistica = NonlinearModelFit[totaleCasi, logistica[t], {M, β, α}, t];
Normal[nlmLogistica]

Out[*]= 18812.4
1 + 59.9171 e-0.106763 t
```

Popolazione limite del modello

```
In[*]:= nlmParametri = nlmLogistica["BestFitParameters"];
popolazioneLimite = Round[M /. nlmParametri]

Out[*]= 18812
```

```

In[ ]:= logisticaPlotLineare = Plot[Normal[nlmLogistica], {t, 1, giorniDal24}, PlotStyle -> {Orange, Thickness[0.015]}, PlotLegends -> Placed[{"funzione logistica"}, {.68, .2}]];
(* *)
fig14 = Show[logisticaPlotLineare, plotLineareDati[ {.7, .1}], Graphics[{Purple, Thick, Dashed, Line[{0, popolazioneLimite}, {giorniDal24, popolazioneLimite}]}],
  PlotRange -> All, PlotLabel -> Style["Veneto: totale giornaliero casi positivi\ne modello logistico", Bold],
  Epilog -> {Inset["linea di stabilizzazione\n contagi = " <> ToString[popolazioneLimite], {5, popolazioneLimite - 2000}, Left], Inset["Aggiornato " <> dataUltimoAggiornamento, Scaled@{.75, .4}]},
  FrameLabel -> {"giorni dal 24 febbraio 2020", "totale positivi (cumulativo)"}, Frame -> True, GridLines -> {griglia, Automatic}]

```

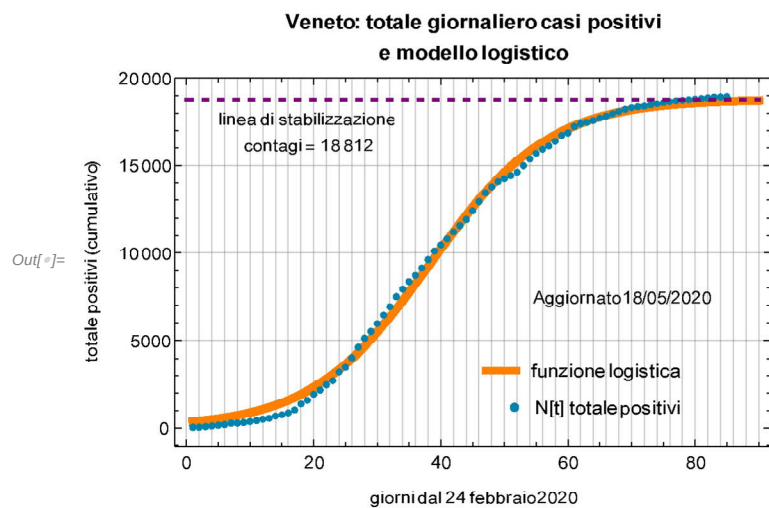


Figura 14

```

In[ ]:= casiCalcolatiLogistica = Table[Normal[nlmLogistica], {t, 1, giorniDal24}];
residui = totaleCasi - Take[casiCalcolatiLogistica, Length[totaleCasi]];
fig15 = ListPlot[residui, PlotLabel -> Style["Residui (mod. logistico): N[t]-n(t)", Bold], Filling -> {1 -> {Axis, {celeste, Lighter[Red, .5]}},
  GridLines -> {griglia, Table[i, {i, Min[residui] - 500, Max[residui] + 500, 100}]}, Frame -> True, FrameLabel -> {"giorni dal 24 febbraio 2020", "residui"},
  PlotRange -> {{0, giorniDal24}, All}, PlotStyle -> {celeste, PointSize[0.012]}, Epilog -> {Inset["Aggiornato " <> dataUltimoAggiornamento, Scaled@{.75, .8}]}]

```

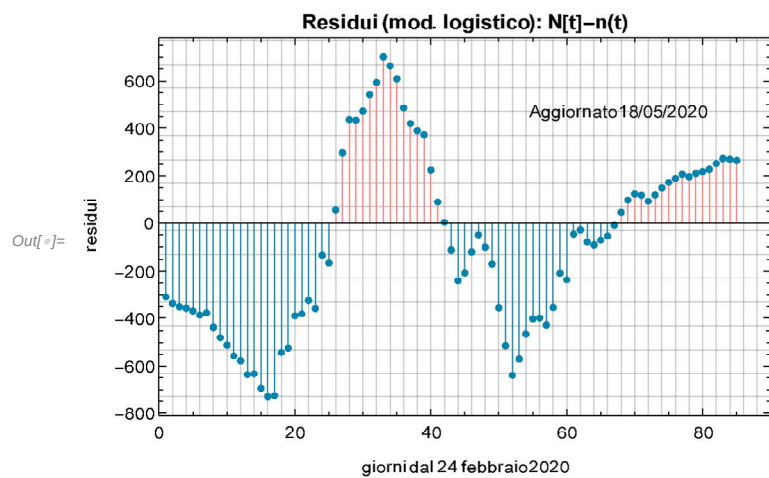


Figura 15

```

In[ ]:= logisticaLogPlot = LogPlot[Normal[nlmLogistica], {t, 1, giorniDal24}, PlotStyle -> {Orange, Thickness[0.01]}, PlotLegends -> Placed[{"funzione logistica"}, {.7, .3}]];
(* *)
asintoto = LogPlot[popolazioneLimite, {t, 0, giorniDal24}, PlotStyle -> {Thick, Dashed, Purple}];
(* *)
fig16 = Show[logisticaLogPlot, asintoto, ListLogPlot[totaleCasi, PlotStyle -> {{celeste, PointSize[Medium]}, {celeste, PointSize[Medium]}}, PlotLegends -> Placed[{"N[t] totale positivi"}, {.715, .2}],
  PlotRange -> {{40, giorniDal24}, {9.2, 9.88}}, GridLines -> {griglia, Automatic}, PlotLabel -> Style["Veneto: totale giornaliero casi positivi\n e modello logistico (logPlot, particolare)", Bold],
  Epilog -> {Inset["linea di stabilizzazione", Scaled@{.05, .82}, Left], Inset["Aggiornato " <> dataUltimoAggiornamento, Scaled@{.75, .5}]},
  FrameLabel -> {"giorni dal 24 febbraio 2020", "totale positivi (cumulativo)"}, Frame -> True, PlotStyle -> {celeste, PointSize[0.012]}}]

```

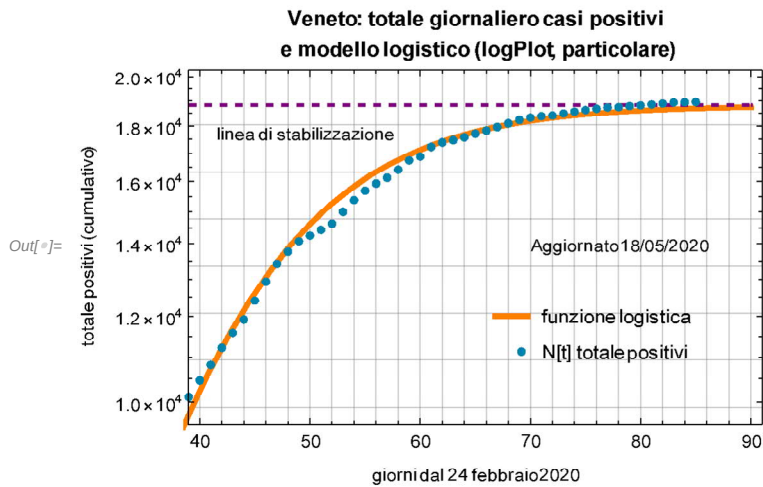


Figura 16

4.3 Variazione giornaliera del contagio

Calcolo dei nuovi positivi giornalieri (gli stessi già presenti nei dati della Protezione Civile e ricavabili dai nuovi_positivi).

```

In[ ]:= nuoviCasi = -Apply[Subtract, Partition[totaleCasi, 2, 1], 1];
In[ ]:= deltaDatiPlotLineare = ListPlot[nuoviCasi, Joined -> True, Filling -> Bottom, Mesh -> All, Frame -> True, PlotLegends -> Placed[{"nuovi contagi"}, {.77, .77}], PlotStyle -> {celeste, PointSize[0.012]};
(* *)
fig17 = Show[deltaDatiPlotLineare, GridLines -> {griglia, Table[i, {i, 0, 600, 100}]}, PlotLabel -> Style["Veneto: variazione giornaliera contagio", Bold, FontSize -> 15], PlotRange -> {{0, giorniDal24}, Automatic},
  Epilog -> {Inset["Aggiornato " <> dataUltimoAggiornamento, Scaled@{.8, .87}]}, Frame -> True, FrameLabel -> {"giorni dal 24 febbraio 2020", "nuovi positivi giornalieri"}]

```

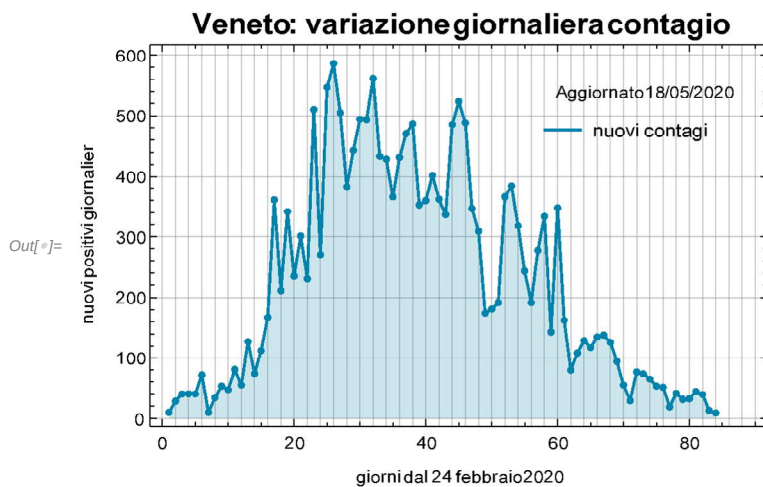


Figura 17

Calcolo simbolico della derivata del modello logistico

```
In[ ]:= derivataLogistica[t_] := D[logistica[t], t];
derivataLogistica[t]
```

$$\text{Out[]} = \frac{e^{-t \alpha} M \alpha \beta}{(1 + e^{-t \alpha} \beta)^2}$$

Il medesimo calcolo ma con i parametri già determinati con l'applicazione dei minimi quadrati

```
In[ ]:= derivNlmLog = D[Normal[nlmLogistica], t]
```

$$\text{Out[]} = \frac{120\,341. e^{-0.106763 t}}{(1 + 59.9171 e^{-0.106763 t})^2}$$

```
In[ ]:= derivataLogisticaPlotLineare = Plot[derivNlmLog, {t, 0, giorniDal24}, PlotStyle -> {Orange, Thickness[0.012]}, PlotLegends -> Placed[{"derivata funzione\nlogistica"}, {.8, .65}]]];
(* *)
```

```
fig18 = Show[derivataLogisticaPlotLineare, deltaDatiPlotLineare, PlotLabel -> Style["Veneto: variazione giornaliera contagio\n e curva derivata", Bold, FontSize -> 15],
PlotRange -> All, Epilog -> {Inset["Aggiornato " <> dataUltimoAggiornamento, Scaled@{.8, .87}]}, Frame -> True,
FrameLabel -> {"giorni dal 24 febbraio 2020", "nuovi positivi giornalieri"}, GridLines -> {griglia, Table[i, {i, 0, 600, 100}]]}]
```

**Veneto: variazione giornaliera contagio
e curva derivata**

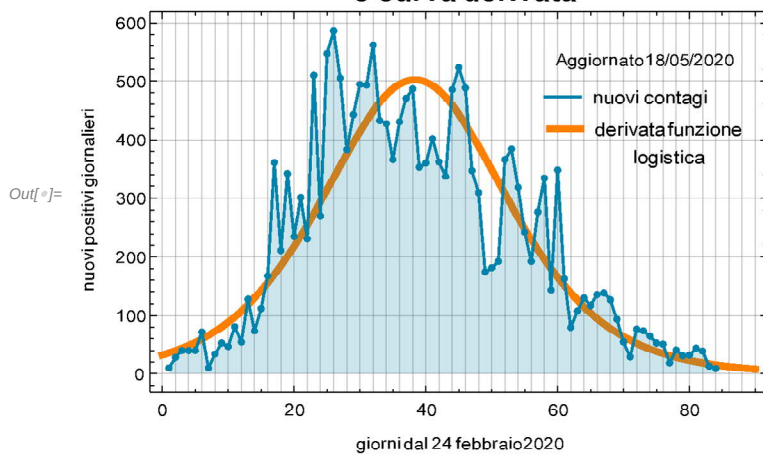


Figura 18

Ricerca del giorno del massimo

```
In[ ]:= derivataSeconda = D[derivNlmLog, t] // Simplify;
giornoDelPicco = Round[t /. NSolve[derivataSeconda == 0, t, Reals]][[1]]
```

Out[]:= 38

e della relativa data

```
In[ ]:= DatePlus[DateObject[{2020, 2, 24}], giornoDelPicco]
```

Out[]:=  Day: **Thu 2 Apr 2020**

Ricerca della condizione asintotica di un solo infetto

```
In[ ]:= dayUnSoloInfetto = Round[t /. NSolve[derivNlmLog == 1, t, Reals]][[2]]]
```

Out[]:= 110

e della relativa data

```
In[ ]:= DatePlus[DateObject[{2020, 2, 24}], dayUnSoloInfetto]
```

```
Out[ ]:=  Day: Sat 13 Jun 2020
```

5. Approfondimenti: il modello SIR (regione Veneto)

Si assegnano i valori ai parametri R_0 e τ .

```
In[ ]:= Clear["r0", "tau"];
tau = 14;
r0 = 4;
i0 = 1 / 250000;
gamma = 1 / tau;
beta = r0 / tau;
{i0, beta, gamma} // N
```

```
Out[ ]:= {4. × 10-6, 0.285714, 0.0714286}
```

Si definiscono le due equazioni differenziali

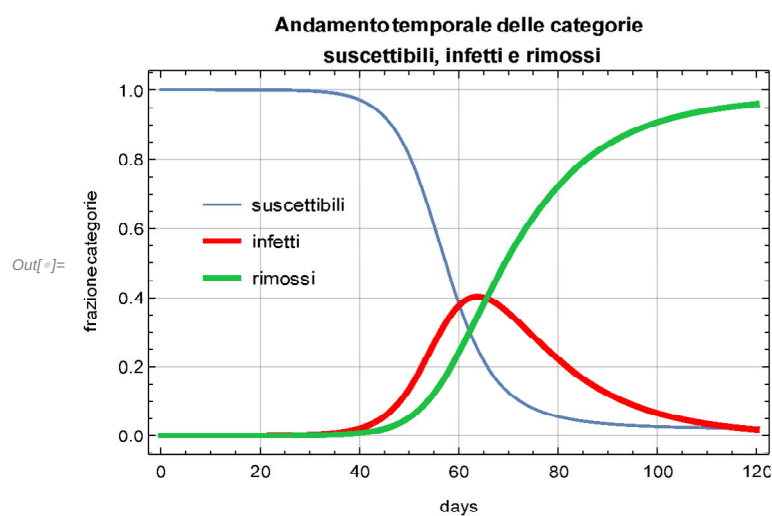
```
In[ ]:= eq1 = s'[t] == -beta s[t] × i[t];
eq2 = i'[t] == beta s[t] × i[t] - gamma i[t];
```

e se ne trovano le soluzioni numeriche nell'intervallo $[0, 120]$.

```
In[ ]:= soln = NDSolve[{eq1, eq2, i[0] == i0, s[0] == 1 - i[0]}, {i, s}, {t, 120}];
```

e le si rappresenta graficamente

```
In[ ]:= fig20 = teoriaSIR = Plot[Evaluate[{s[t], i[t], 1 - s[t] - i[t]} /. soln], {t, 0, 120}, PlotRange → All,
PlotStyle → {Automatic, {Red, Thickness[0.01]}, {verde, Thickness[0.01]}}, PlotLegends → Placed[{"suscettibili", "infetti", "rimossi"}, {.2, .55}], GridLines → Automatic,
Frame → True, FrameLabel → {"days", "frazione categorie"}, PlotLabel → Style["Andamento temporale delle categorie\n suscettibili, infetti e rimossi", Bold]]
```



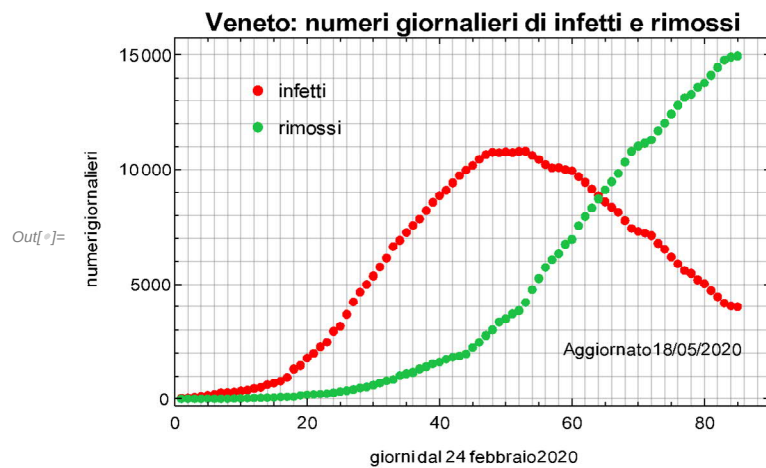
Estrae i dati di interesse della regione

```
In[ ]:= dimessiGuariti = estraeDati["dimessi_guariti", datiRegione];
deceduti = estraeDati["deceduti", datiRegione];
tamponi = estraeDati["tamponi", datiRegione];
```


Elaborazione e grafico dell'andamento degli infetti e dei risolti

```
In[*]:= infetti = totaleCasi - dimessiGuariti - deceduti;
risolti = dimessiGuariti + deceduti;
```

```
In[*]:= fig21 = ListPlot[{infetti, risolti}, PlotStyle -> {{Red, PointSize[0.015]}, {verde, PointSize[0.015]}}, GridLines -> {griglia, Table[i, {i, 0, 15000, 1000}]},
PlotRange -> {{0, giorniDal24}, Automatic}, Frame -> True, FrameLabel -> {"giorni dal 24 febbraio 2020", "numeri giornalieri"}, PlotLegends -> Placed[{"infetti", "rimossi"}, {.2, .8}],
PlotLabel -> Style["Veneto: numeri giornalieri di infetti e rimossi", Bold, FontSize -> 13], Epilog -> {Inset["Aggiornato " <> dataUltimoAggiornamento, Scaled@{.8, .15}]}}]
```



Dati Italia

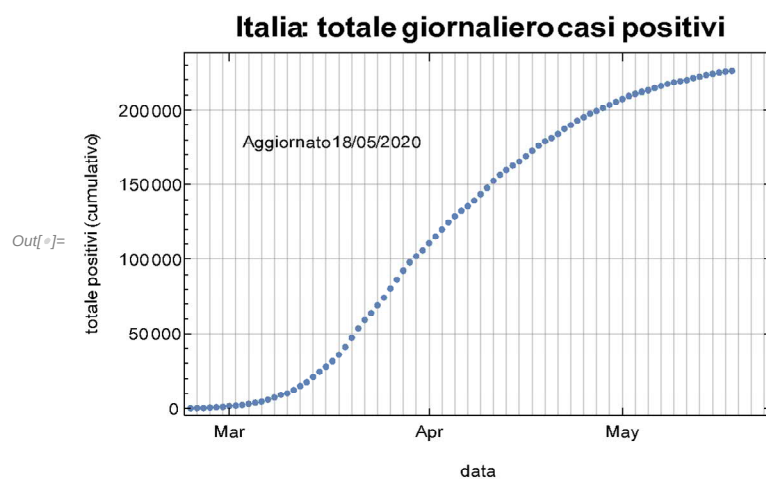
Qui si ripercorre quanto fatto per la regione precedente trattando i dati nazionali

```
In[*]:= (* Select elimina l'eventuale ACapo finale *)
datiIT = Select[Import["datiNazionali" <> formatoISO <> ".csv"], # & # {} &];;
```

```
In[*]:= totaleCasiIT = Drop[Transpose[datiIT][[12]], 1];
```

```
In[*]:= datiPlotLineareIT[posizioneRelativa_List] := ListPlot[totaleCasiIT, PlotStyle -> {celeste, PointSize[0.012]}, PlotLegends -> Placed[{"N[t] totale positivi"}, posizioneRelativa]];
```

```
In[*]:= DateListPlot[totaleCasiIT, {2020, 2, 24}, PlotRange -> {{2020, 2, 23}, dataDal24}, Automatic},
GridLines -> {Table[{2020, 2, 23 + i}, {i, 0, giorniDal24, 2}], Automatic}, PlotLabel -> Style["Italia: totale giornaliero casi positivi", Bold, FontSize -> 15],
Joined -> False, Epilog -> Inset["Aggiornato " <> dataUltimoAggiornamento, Scaled@{.25, .75}], FrameLabel -> {"data", "totale positivi (cumulativo)"}]
```



Calcolo numerico della funzione logistica ottimale


```
In[ ]:= logistica[t_] := M / (1 +  $\beta$  Exp[- $\alpha$  t]);
nlmLogisticaIT = NonlinearModelFit[totaleCasiIT, logistica[t], {M,  $\beta$ ,  $\alpha$ }, t];
Normal[nlmLogisticaIT]
```

```
Out[ ]:= 
$$\frac{220777.}{1 + 49.242 e^{-0.0991589 t}}$$

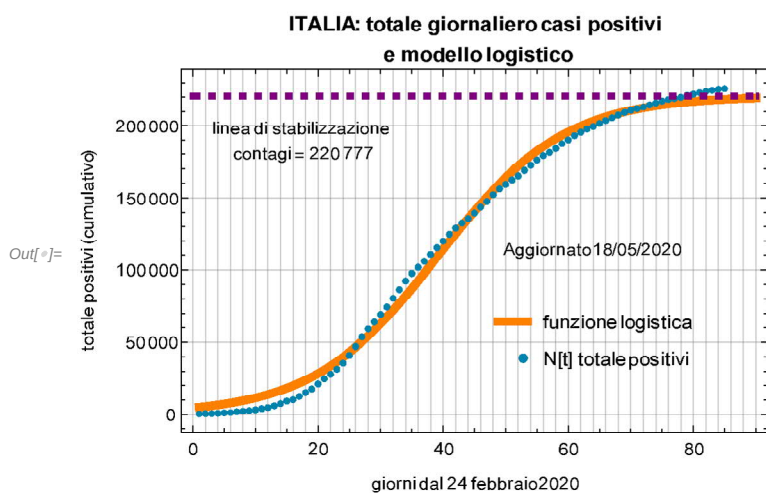
```

i relativi parametri sono

```
In[ ]:= nlmParametriIT = nlmLogisticaIT["BestFitParameters"]
popolazioneLimiteIT = Round[M /. nlmParametriIT];
```

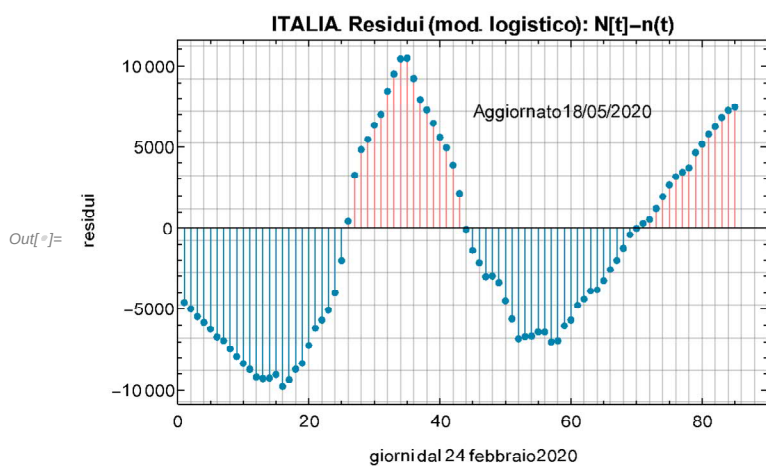
```
Out[ ]:= {M → 220777.,  $\beta$  → 49.242,  $\alpha$  → 0.0991589}
```

```
In[ ]:= logisticaPlotLineareIT = Plot[Normal[nlmLogisticaIT], {t, 1, giorniDal24}, PlotStyle → {Orange, Thickness[0.015]}, PlotLegends → Placed[{"funzione logistica"}, {.7, .3}]];
Show[logisticaPlotLineareIT, datiPlotLineareIT[.715, .2]], Graphics[{Purple, Thickness[0.012], Dashed, Line[{0, popolazioneLimiteIT}, {giorniDal24, popolazioneLimiteIT}]}]],
PlotRange -> All, PlotLabel → Style["ITALIA: totale giornaliero casi positivi\n e modello logistico", Bold],
Epilog → {Inset["linea di stabilizzazione\n contagi = " <> ToString[popolazioneLimiteIT], Scaled@{.2, .8}], Inset["Aggiornato " <> dataUltimoAggiornamento, Scaled@{.7, .5}]},
FrameLabel → {"giorni dal 24 febbraio 2020", "totale positivi (cumulativo)", Frame -> True, GridLines → {griglia, Automatic}}
```



```
In[ ]:= casiCalcolatiLogisticaIT = Table[Normal[nlmLogisticaIT], {t, 1, giorniDal24}];
```

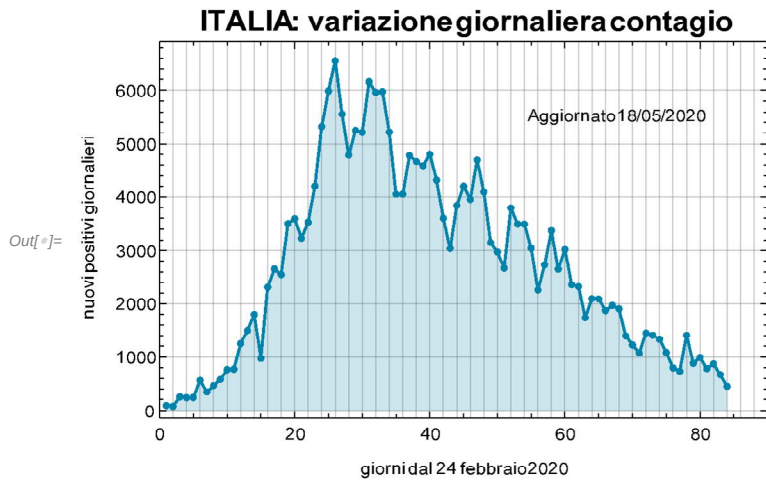
```
In[ ]:= residuiIT = totaleCasiIT - Take[casiCalcolatiLogisticaIT, Length[totaleCasiIT]];
ListPlot[residuiIT, PlotLabel → Style["ITALIA. Residui (mod. logistico): N[t]-n(t)", Bold], Filling → {1 → {Axis, {celeste, Lighter[Red, .5]}},
GridLines → {griglia, Table[i, {i, Min[residuiIT] - 1000, Max[residuiIT] + 1000, 2000}]}, Frame -> True, FrameLabel → {"giorni dal 24 febbraio 2020", "residui"},
PlotRange → {{0, giorniDal24}, All}, PlotStyle → {celeste, PointSize[0.012]}, Epilog → {Inset["Aggiornato " <> dataUltimoAggiornamento, Scaled@{.65, .8}]}}
```



Calcolo delle variazioni giornaliere dei positivi

```
In[*]:= nuoviCasiIT = -Apply[Subtract, Partition[totaleCasiIT, 2, 1], 1];
```

```
In[*]:= ListPlot[nuoviCasiIT, Joined → True, Filling → Bottom, Mesh → All, GridLines → {griglia, Table[i, {i, 0, 7000, 1000}]},
  PlotLabel → Style["ITALIA: variazione giornaliera contagio", Bold, FontSize → 15], PlotRange → {{0, giorniDal24}, Automatic}, Epilog → {Inset["Aggiornato " <> dataUltimoAggiornamento, Scaled@{.75, .8}]},
  Frame -> True, FrameLabel → {"giorni dal 24 febbraio 2020", "nuovi positivi giornalieri"}, PlotStyle → {celeste, PointSize[0.012]}]
```



Calcolo della derivata della funzione logistica per l'intera nazione

```
In[*]:= derivNlmLogIT = D[Normal[nlmLogisticaIT], t]
```

$$\text{Out[*]} = \frac{1.07801 \times 10^6 e^{-0.0991589 t}}{(1 + 49.242 e^{-0.0991589 t})^2}$$

```
In[*]:= derivataLogisticaPlotLineareIT = Plot[derivNlmLogIT, {t, 0, 90}, PlotLegends → Placed[{"funzione derivata"}, {.8, .65}], PlotStyle → {Orange, Thickness[0.012]};
deltaDatiPlotLineareIT = ListPlot[nuoviCasiIT, Joined → True, Filling → Bottom, Mesh → All, Frame -> True, PlotLegends → Placed[{"nuovi contagi"}, {.77, .77}], PlotStyle → {celeste, PointSize[0.012]}];
(* *)
Show[derivataLogisticaPlotLineareIT, deltaDatiPlotLineareIT, PlotLabel → Style["ITALIA: variazione giornaliera contagio\n e curva derivata", Bold, FontSize → 15],
  PlotRange → All, Epilog → {Inset["Aggiornato " <> dataUltimoAggiornamento, Scaled@{.8, .87}]}, Frame -> True,
  FrameLabel → {"giorni dal 24 febbraio 2020", "nuovi positivi giornalieri"}, GridLines → {griglia, Table[i, {i, 0, 7000, 1000}]}
```

