Georgia State University | SCHOOL OF PUBLIC HEALTH

MCMC I for Infectious Diseases

Lab & Activity

Resources

Presentation Activity

PollEv.com/amandableichrodt759

Toolbox Code

https://github.com/gchowell/paramEstimation_forecasting_ODEmodels

Toolbox Tutorial

https://pubmed.ncbi.nlm.nih.gov/38378161/

Emory’s Apporto

https://rsphemory.apporto.com/

apporto

HELP

EMORY | ROLLINS SCHOOL OF PUBLIC HEALTH

Welcome to your Emory RSPH virtual desktop.

Email or Username *

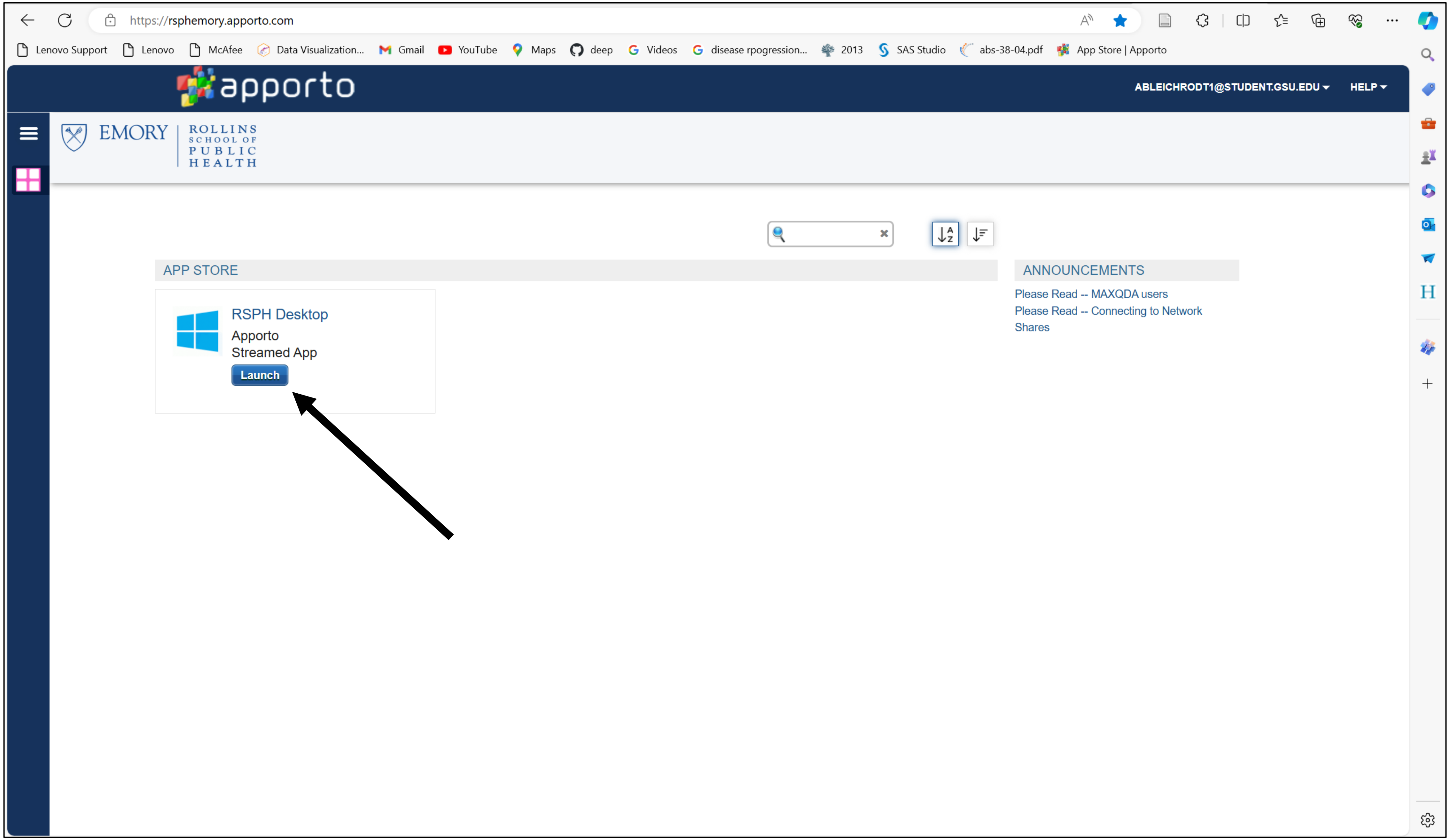
Password *

I'm not a robot

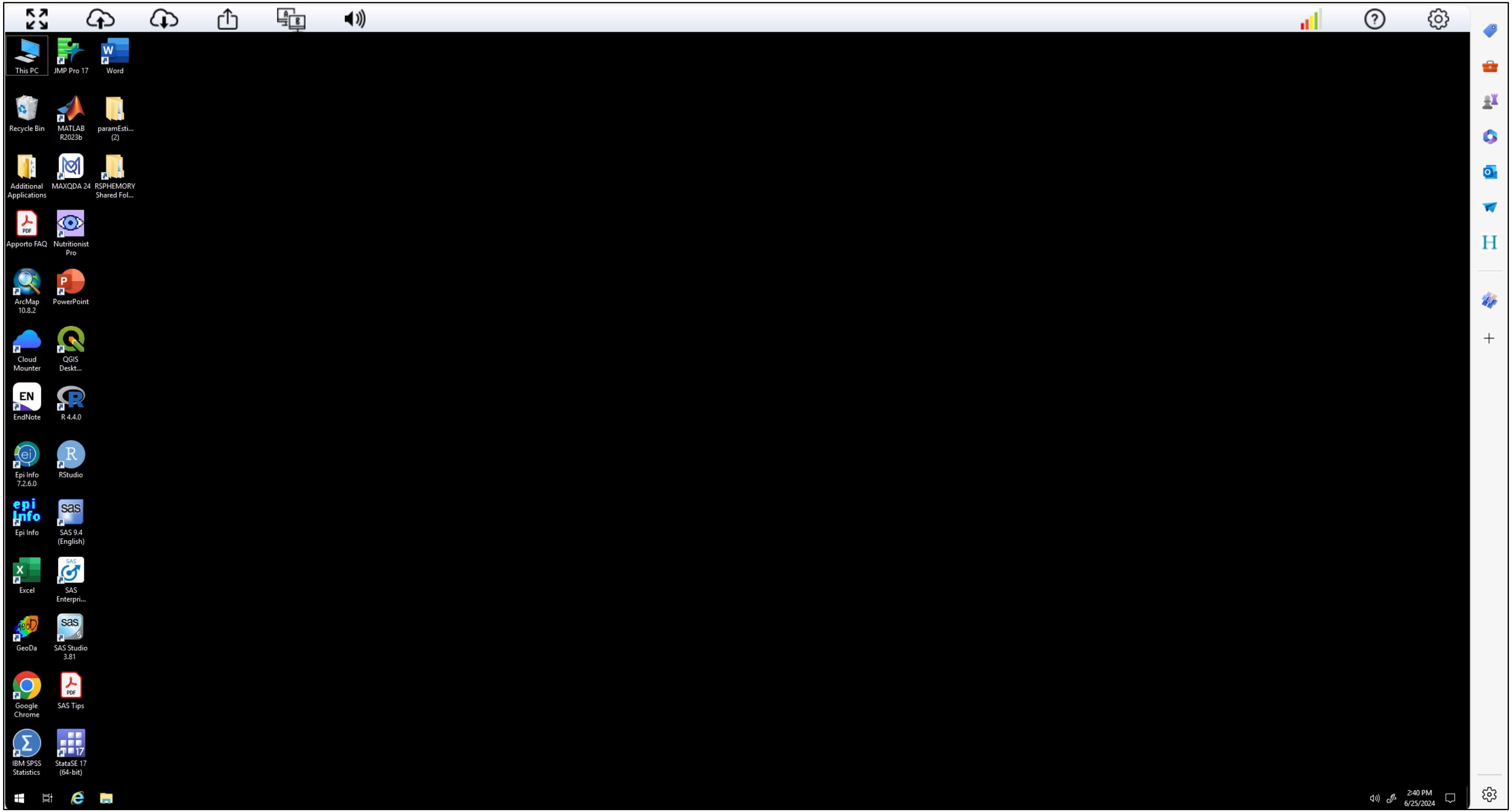
reCAPTCHA
Privacy - Terms

Log in

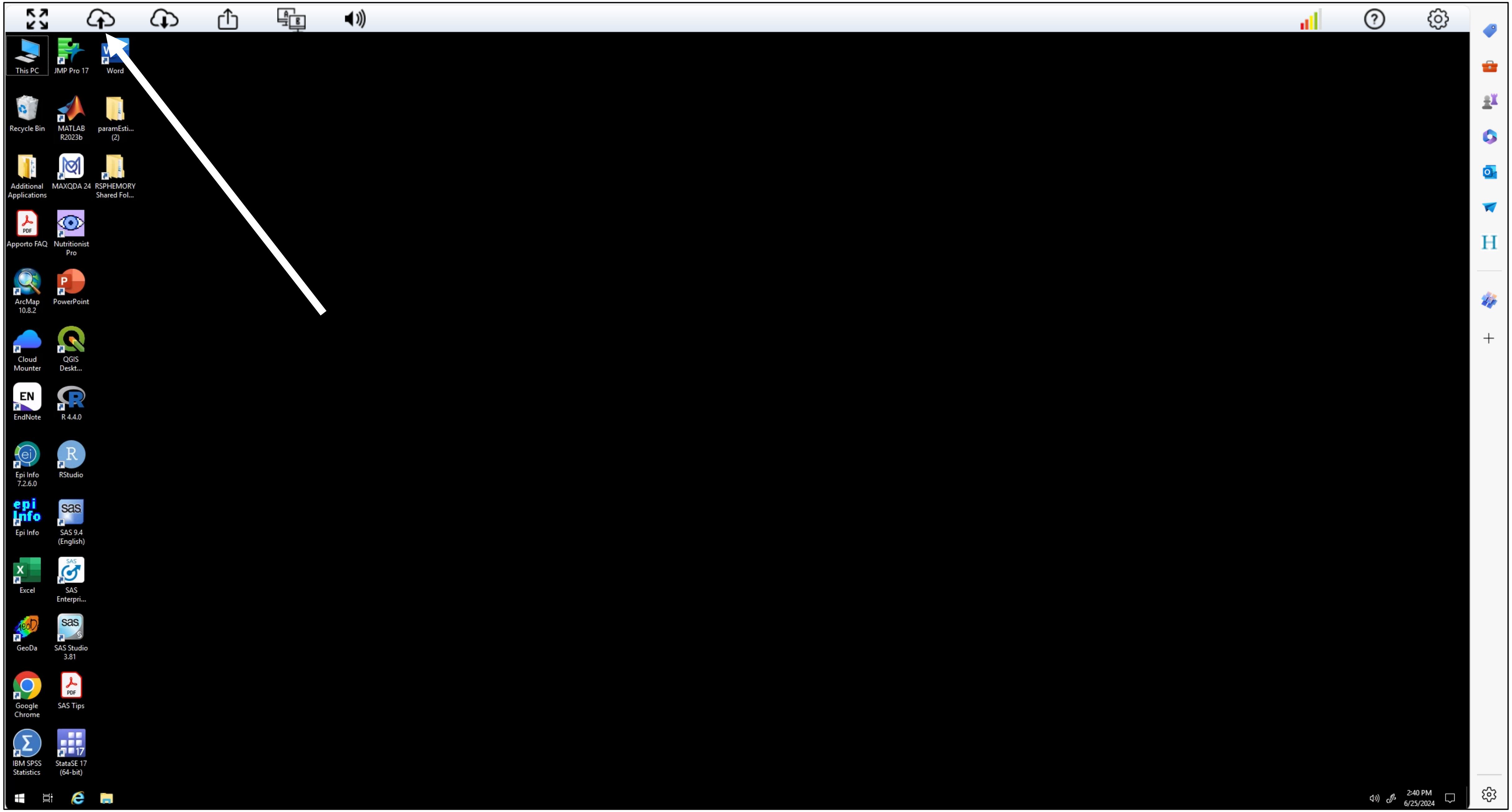
Forgot your password?

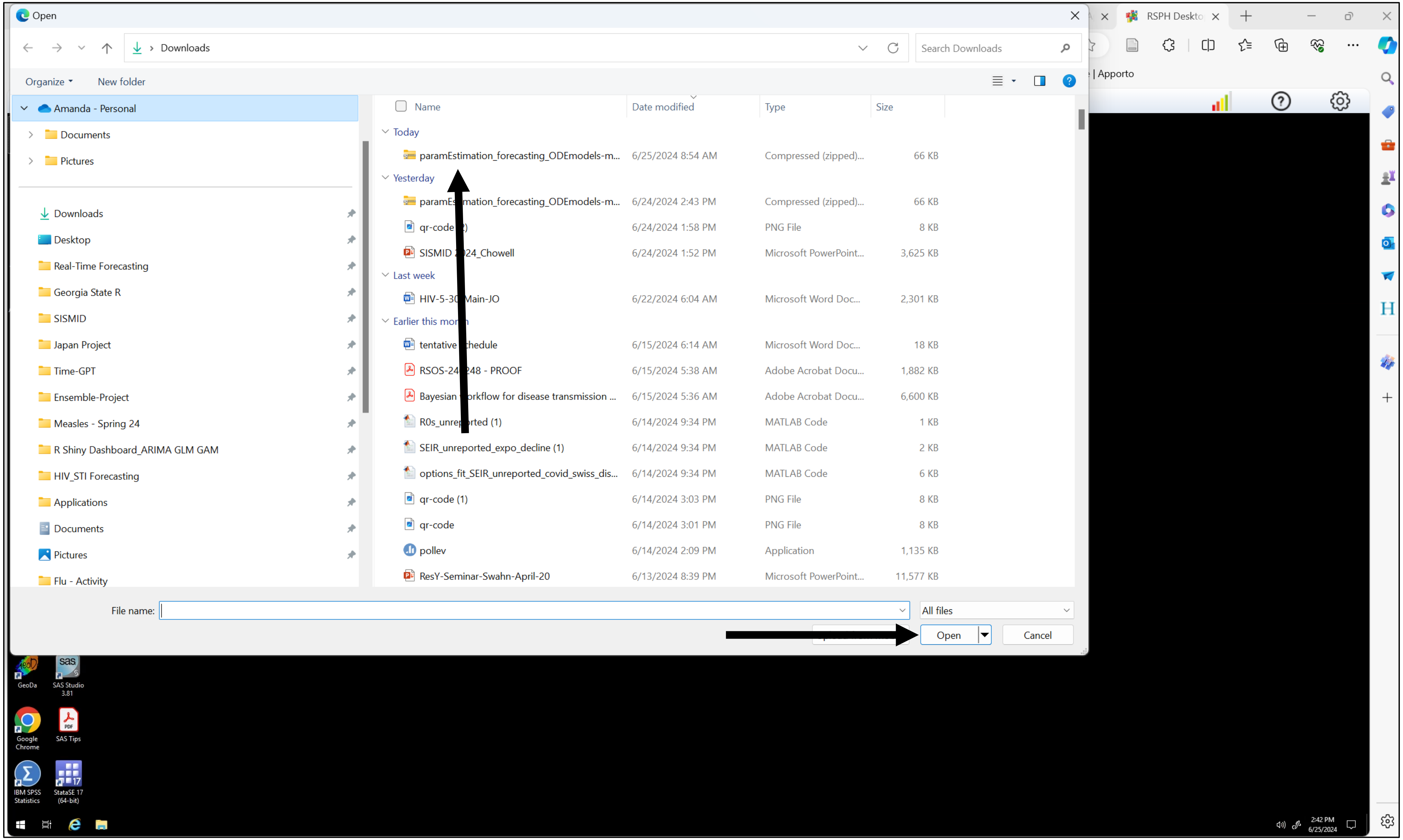


https://rsphemory.apporto.com
Lenovo Support
Lenovo
McAfee
Data Visualization...
Gmail
YouTube
Maps
deep
Videos
disease rpogression...
2013
SAS Studio
abs-38-04.pdf
App Store | Apporto
apporto
ABLEICHRODT1@STUDENT.GSU.EDU
HELP
EMORY
ROLLINS SCHOOL OF PUBLIC HEALTH
APP STORE
RSPH Desktop
Apporto
Streamed App
Launch
ANNOUNCEMENTS
Please Read -- MAXQDA users
Please Read -- Connecting to Network Shares

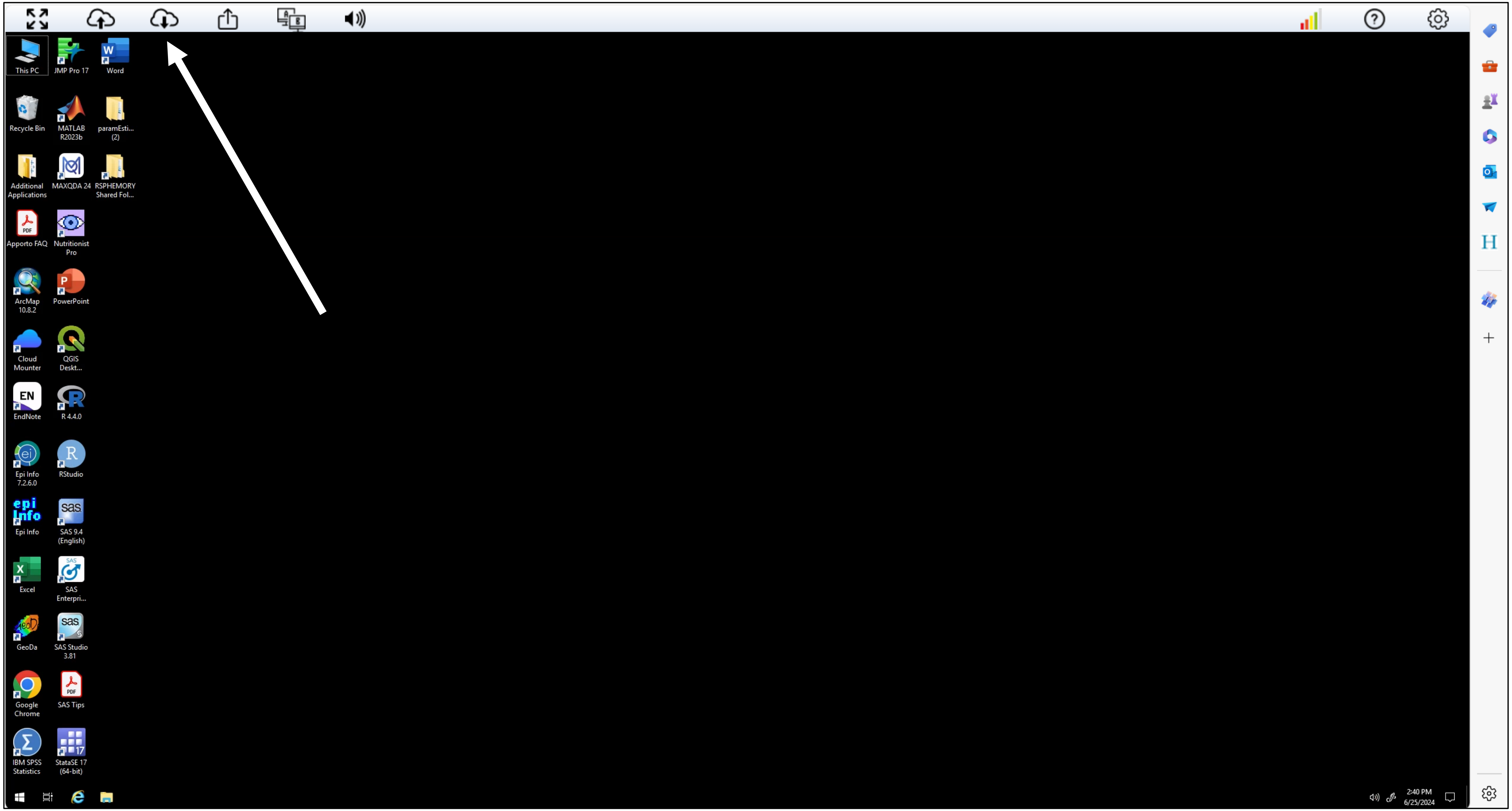

This PC
JMP Pro 17
Word
Recycle Bin
MATLAB R2023b
paramEsti... (2)
Additional Applications
MAXQDA 24
RSPHEMORY Shared Fol...
Apporto FAQ
Nutritionist Pro
ArcMap 10.8.2
PowerPoint
Cloud Mounter
QGIS Deskt...
EndNote
R 4.4.0
Epi Info 7.2.6.0
RStudio
Epi Info
SAS 9.4 (English)
Excel
SAS Enterpri...
GeoDa
SAS Studio 3.81
Google Chrome
SAS Tips
IBM SPSS Statistics
StataSE 17 (64-bit)
2:40 PM
6/25/2024

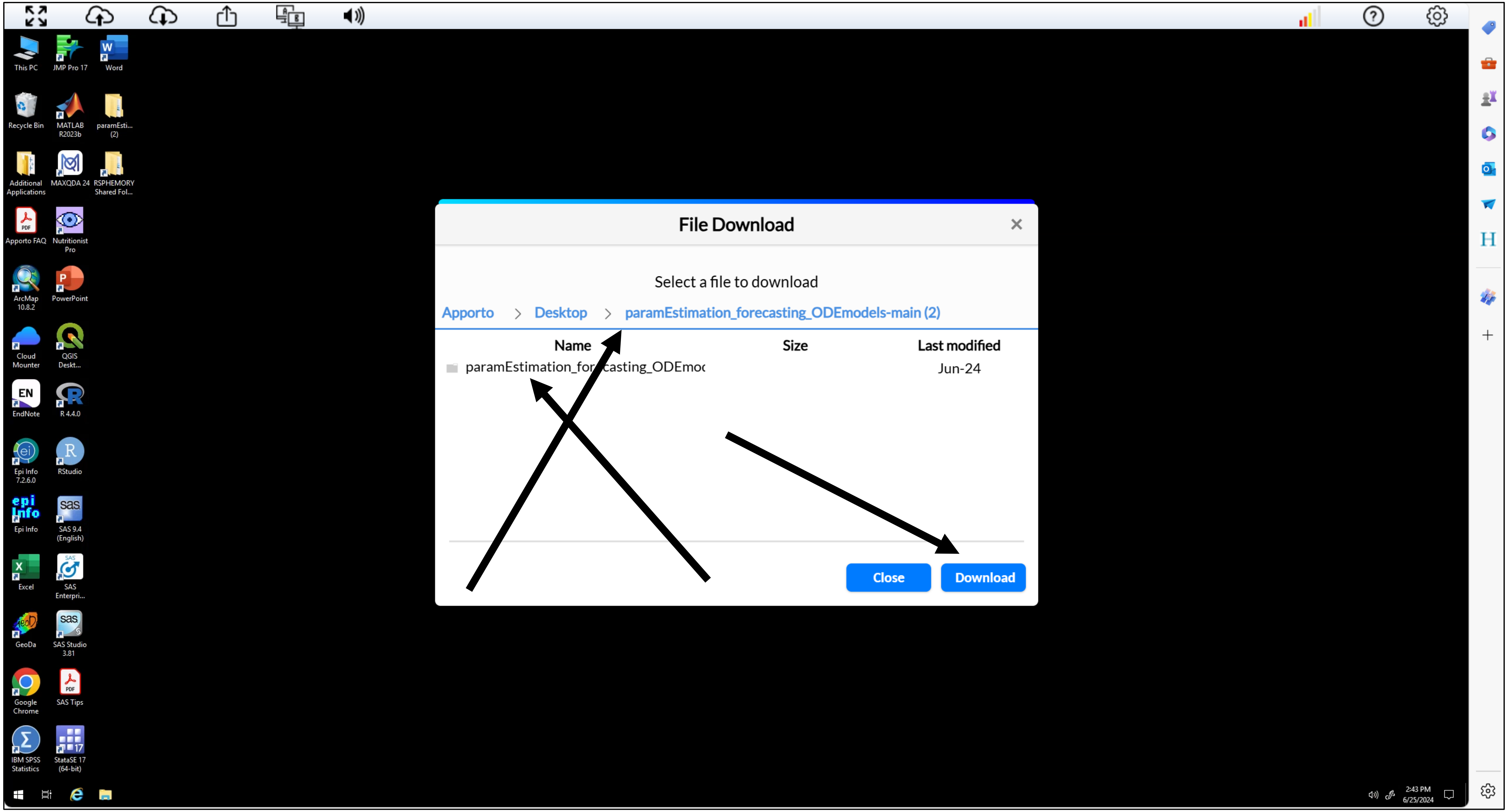
Uploading Materials


This PC
JMP Pro 17
Word
Recycle Bin
MATLAB R2023b
paramEsti... (2)
Additional Applications
MAXQDA 24
RSPHEMORY Shared Fol...
Apporto FAQ
Nutritionist Pro
ArcMap 10.8.2
PowerPoint
Cloud Mounter
QGIS Deskt...
EndNote
R 4.4.0
Epi Info 7.2.6.0
RStudio
Epi Info
SAS 9.4 (English)
Excel
SAS Enterpri...
GeoDa
SAS Studio 3.81
Google Chrome
SAS Tips
IBM SPSS Statistics
StataSE 17 (64-bit)
2:40 PM
6/25/2024

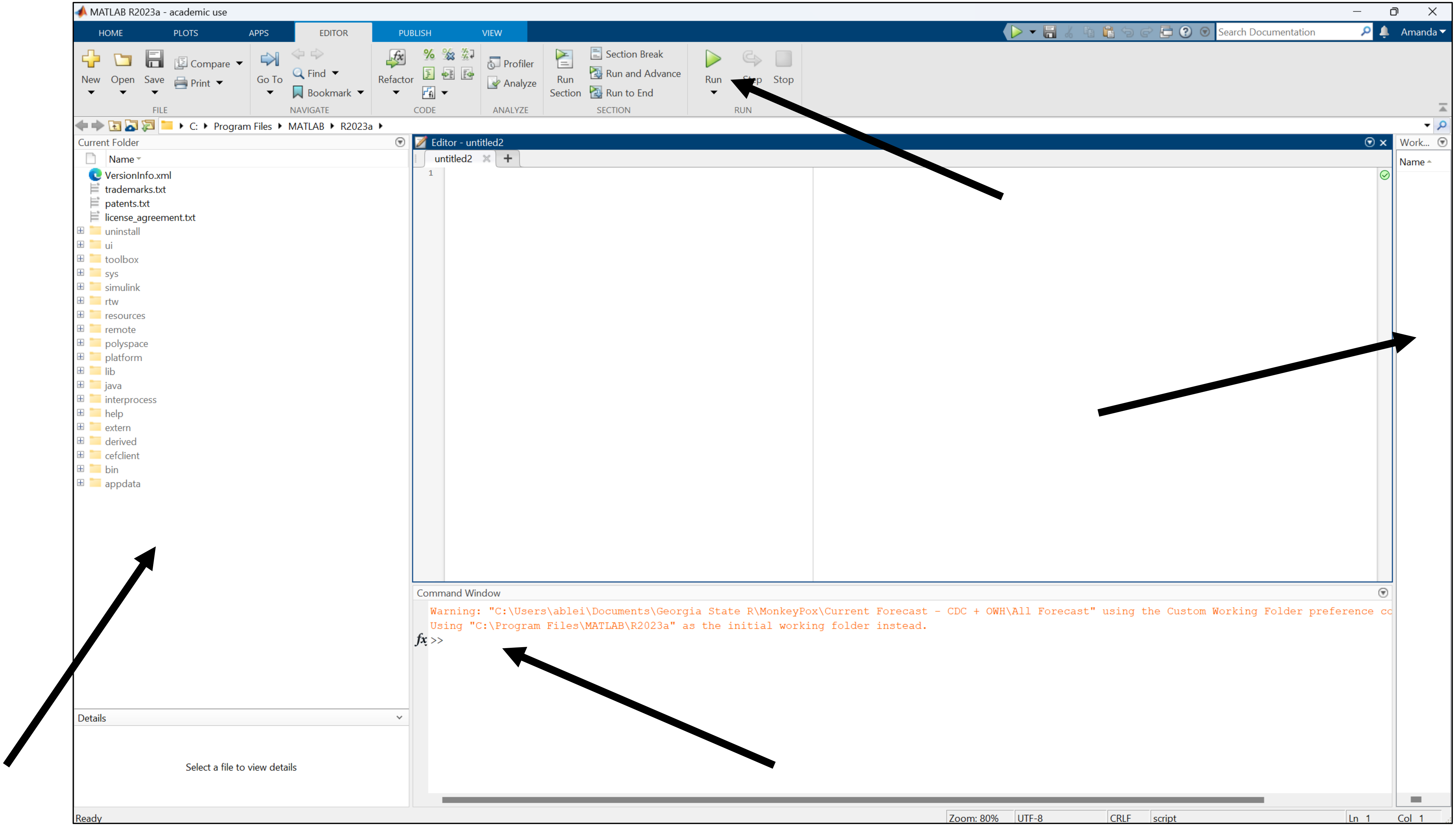

Open
Downloads
Search Downloads
Organize
New folder
Amanda - Personal
Documents
Pictures
Downloads
Desktop
Real-Time Forecasting
Georgia State R
SISMID
Japan Project
Time-GPT
Ensemble-Project
Measles - Spring 24
R Shiny Dashboard_ARIMA GLM GAM
HIV_STI Forecasting
Applications
Documents
Pictures
Flu - Activity
Name
Date modified
Type
Size
Today
paramEstimation_forecasting_ODEmodels-m...
6/25/2024 8:54 AM
Compressed (zipped)...
66 KB
Yesterday
6/24/2024 2:43 PM
6/24/2024 1:58 PM
PNG File
8 KB
6/24/2024 1:52 PM
Microsoft PowerPoint...
3,625 KB
Last week
6/22/2024 6:04 AM
Microsoft Word Doc...
2,301 KB
Earlier this mo
6/15/2024 6:14 AM
18 KB
6/15/2024 5:38 AM
Adobe Acrobat Docu...
1,882 KB
6/15/2024 5:36 AM
6,600 KB
6/14/2024 9:34 PM
MATLAB Code
1 KB
SEIR_unreported_expo_decline (1)
2 KB
options_fit_SEIR_unreported_covid_swiss_dis...
6 KB
qr-code (1)
6/14/2024 3:03 PM
qr-code
6/14/2024 3:01 PM
pollev
6/14/2024 2:09 PM
Application
1,135 KB
ResY-Seminar-Swahn-April-20
6/13/2024 8:39 PM
11,577 KB
File name:
All files
Open
Cancel
RSPH Deskto
Apporto
GeoDa
SAS Studio 3.81
Google Chrome
SAS Tips
IBM SPSS Statistics
StataSE 17 (64-bit)
2:42 PM
6/25/2024

Downloading Data


This PC
JMP Pro 17
Word
Recycle Bin
MATLAB R2023b
paramEsti... (2)
Additional Applications
MAXQDA 24
RSPHEMORY Shared Fol...
Apporto FAQ
Nutritionist Pro
ArcMap 10.8.2
PowerPoint
Cloud Mounter
QGIS Deskt...
EndNote
R 4.4.0
Epi Info 7.2.6.0
RStudio
Epi Info
SAS 9.4 (English)
Excel
SAS Enterpri...
GeoDa
SAS Studio 3.81
Google Chrome
SAS Tips
IBM SPSS Statistics
StataSE 17 (64-bit)
2:40 PM
6/25/2024


File Download
Select a file to download
Apporto > Desktop > paramEstimation_forecasting_ODEmodels-main (2)
Name
Size
Last modified
paramEstimation_forecasting_ODEmoc
Jun-24
Close
Download
This PC
JMP Pro 17
Word
Recycle Bin
MATLAB R2023b
paramEsti... (2)
Additional Applications
MAXQDA 24
RSPHEMORY Shared Fol...
Apporto FAQ
Nutritionist Pro
ArcMap 10.8.2
PowerPoint
Cloud Mounter
QGIS Deskt...
EndNote
R 4.4.0
Epi Info 7.2.6.0
RStudio
Epi Info
SAS 9.4 (English)
Excel
SAS Enterpri...
GeoDa
SAS Studio 3.81
Google Chrome
SAS Tips
IBM SPSS Statistics
StataSE 17 (64-bit)
2:43 PM
6/25/2024

Exploring MATLAB


MATLAB R2023a - academic use
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Amanda
New
Open
Save
Compare
Print
FILE
Go To
Find
Bookmark
NAVIGATE
Refactor
CODE
Profiler
Analyze
ANALYZE
Run Section
Section Break
Run and Advance
Run to End
SECTION
Run
Step
Stop
RUN
C: ▸ Program Files ▸ MATLAB ▸ R2023a ▸
Current Folder
Name
VersionInfo.xml
trademarks.txt
patents.txt
license_agreement.txt
uninstall
ui
toolbox
sys
simulink
rtw
resources
remote
polyspace
platform
lib
java
interprocess
help
extern
derived
cefclient
bin
appdata
Details
Select a file to view details
Editor - untitled2
untitled2
Work...
Name
Command Window
Warning: "C:\Users\ablei\Documents\Georgia State R\MonkeyPox\Current Forecast - CDC + OWH\All Forecast" using the Custom Working Folder preference co
Using "C:\Program Files\MATLAB\R2023a" as the initial working folder instead.
fx >>
Ready
Zoom: 80%
UTF-8
CRLF
script
Ln 1
Col 1

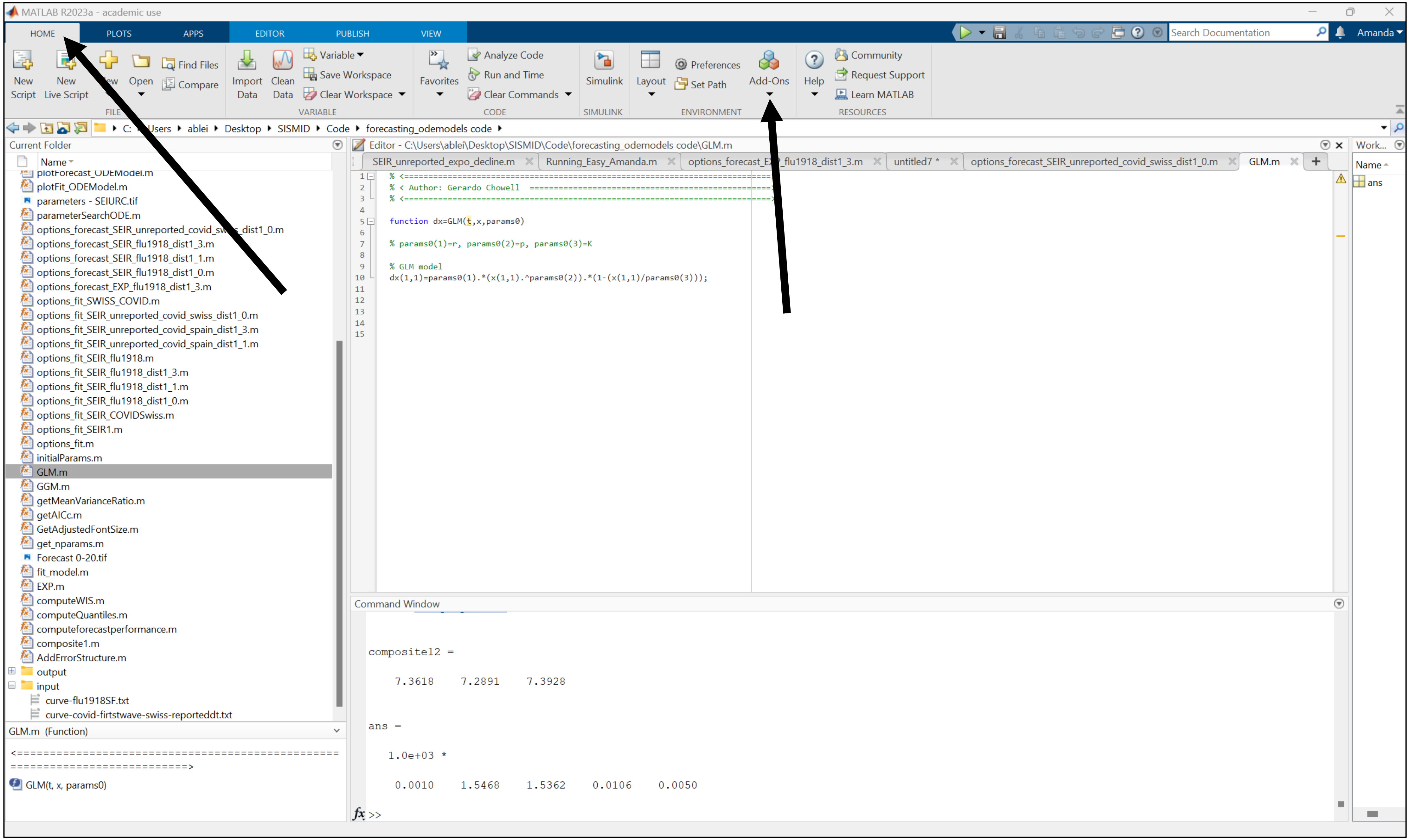
Needed Settings

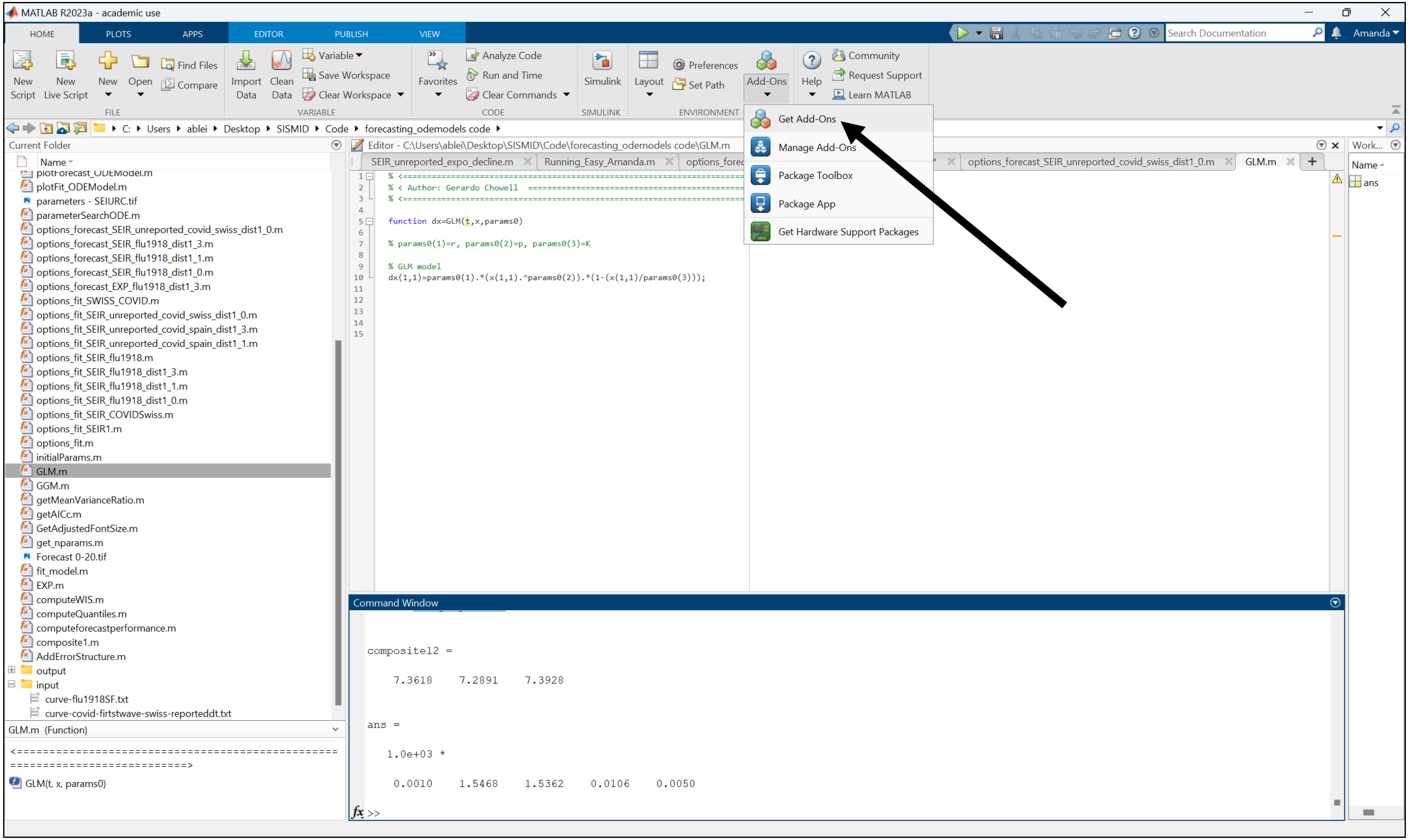
Needed MATLAB Functions
Curve Fitting Toolbox
Global Optimization Toolbox
Optimization Toolbox
Signal Processing Toolbox
Simulink
Statistics and Machine Learning Toolbox

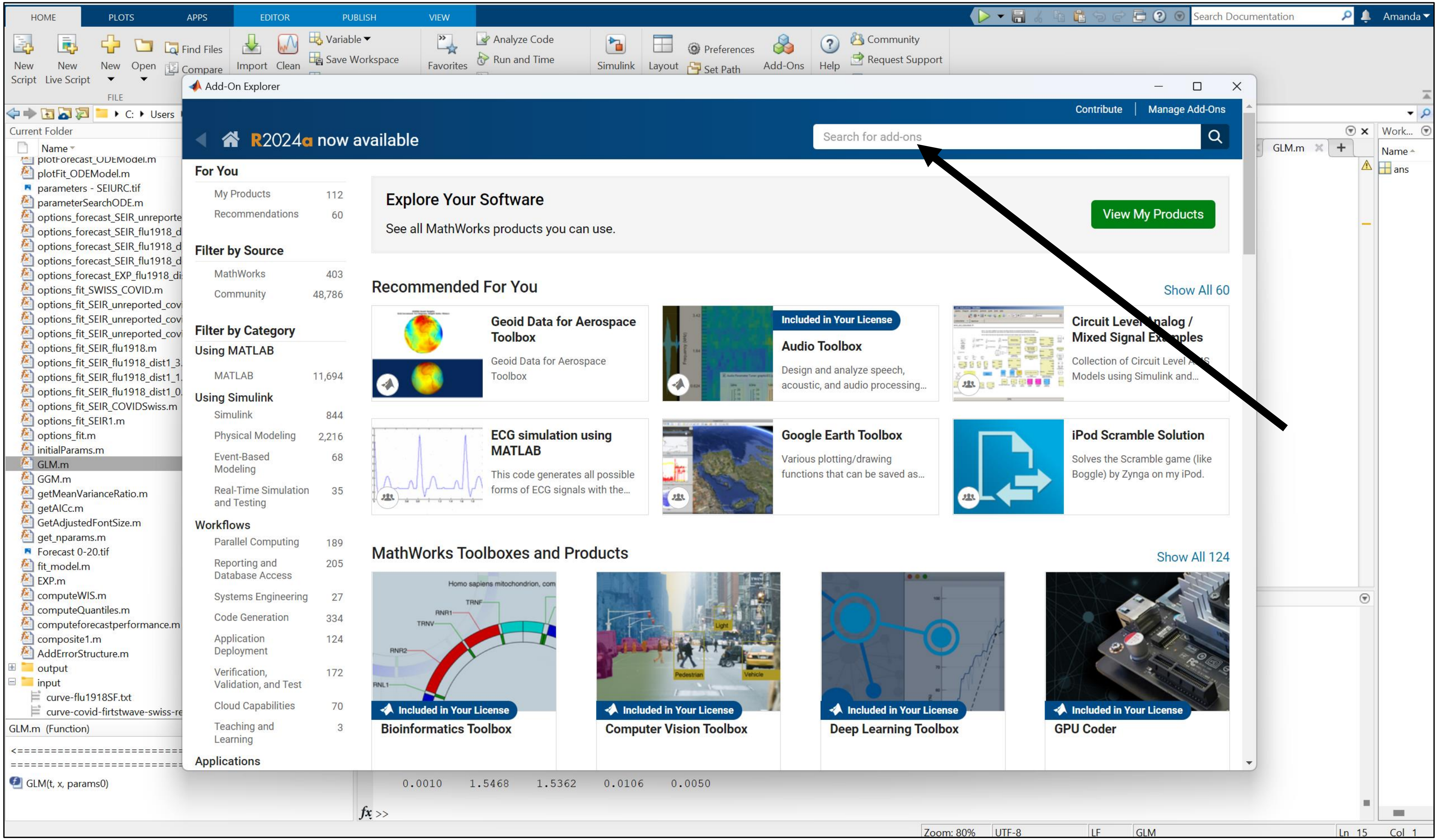
Needed Folders
input
output

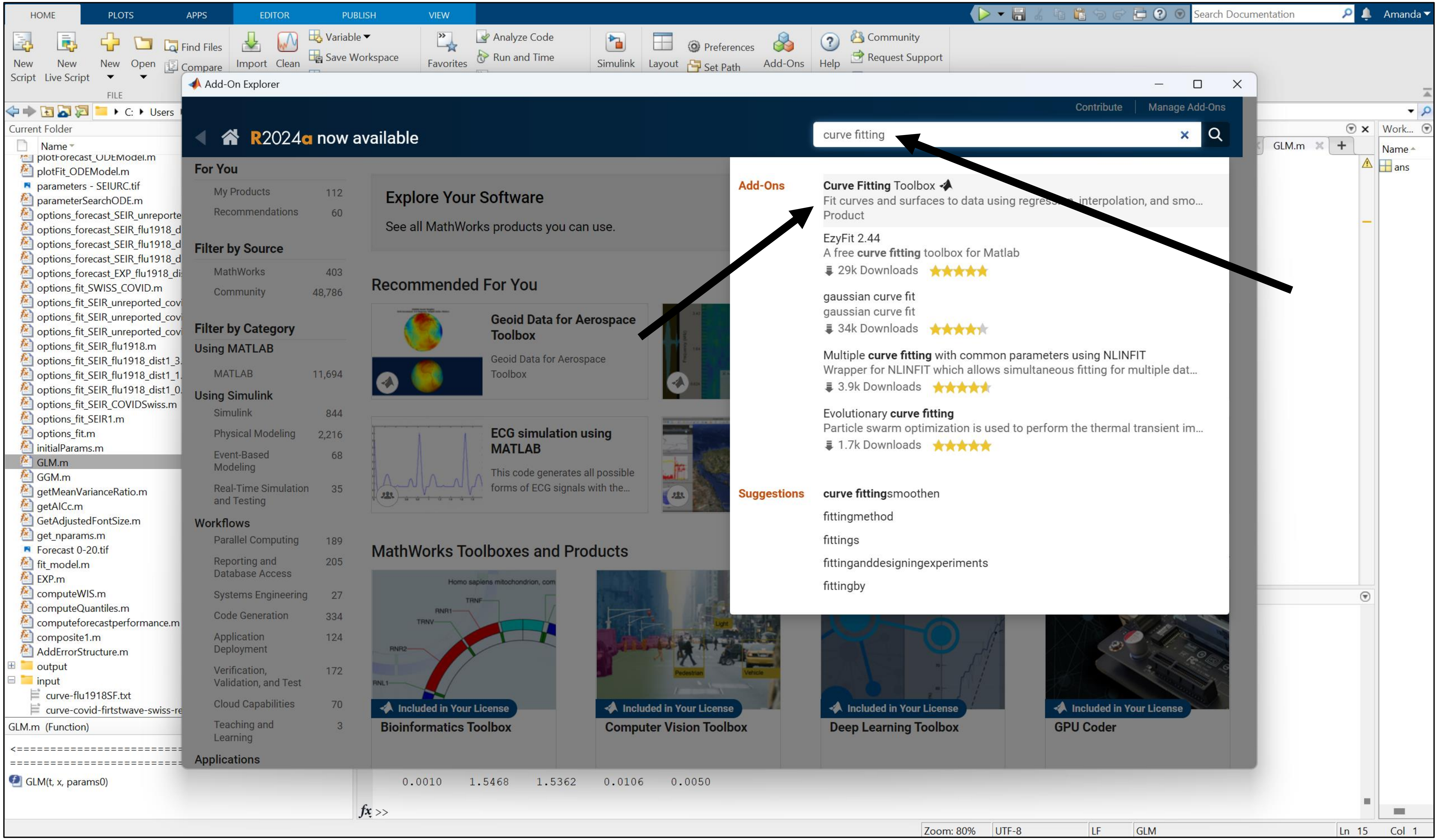
Adding Packages

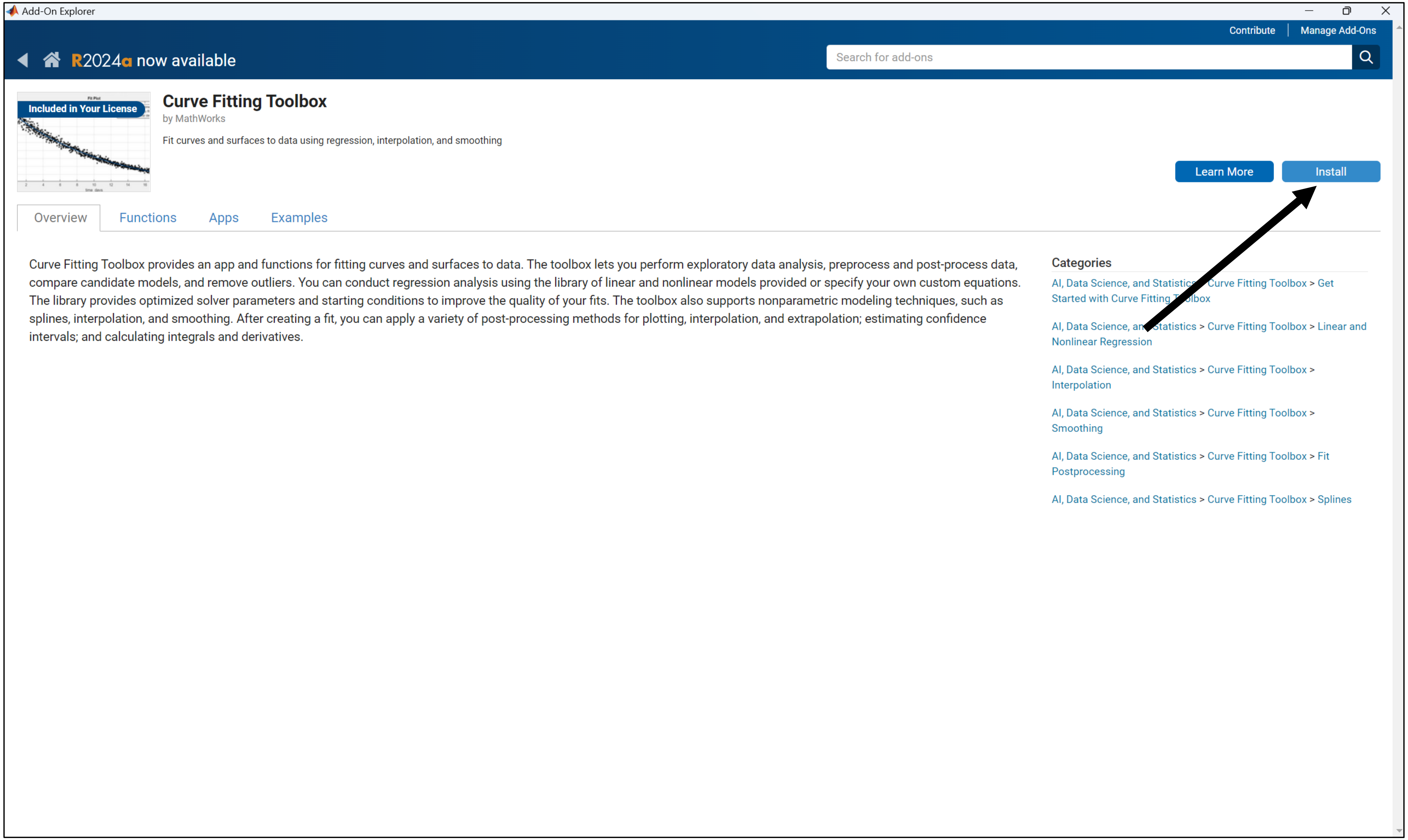
**Not needed if using Emory's Virtual Lab*

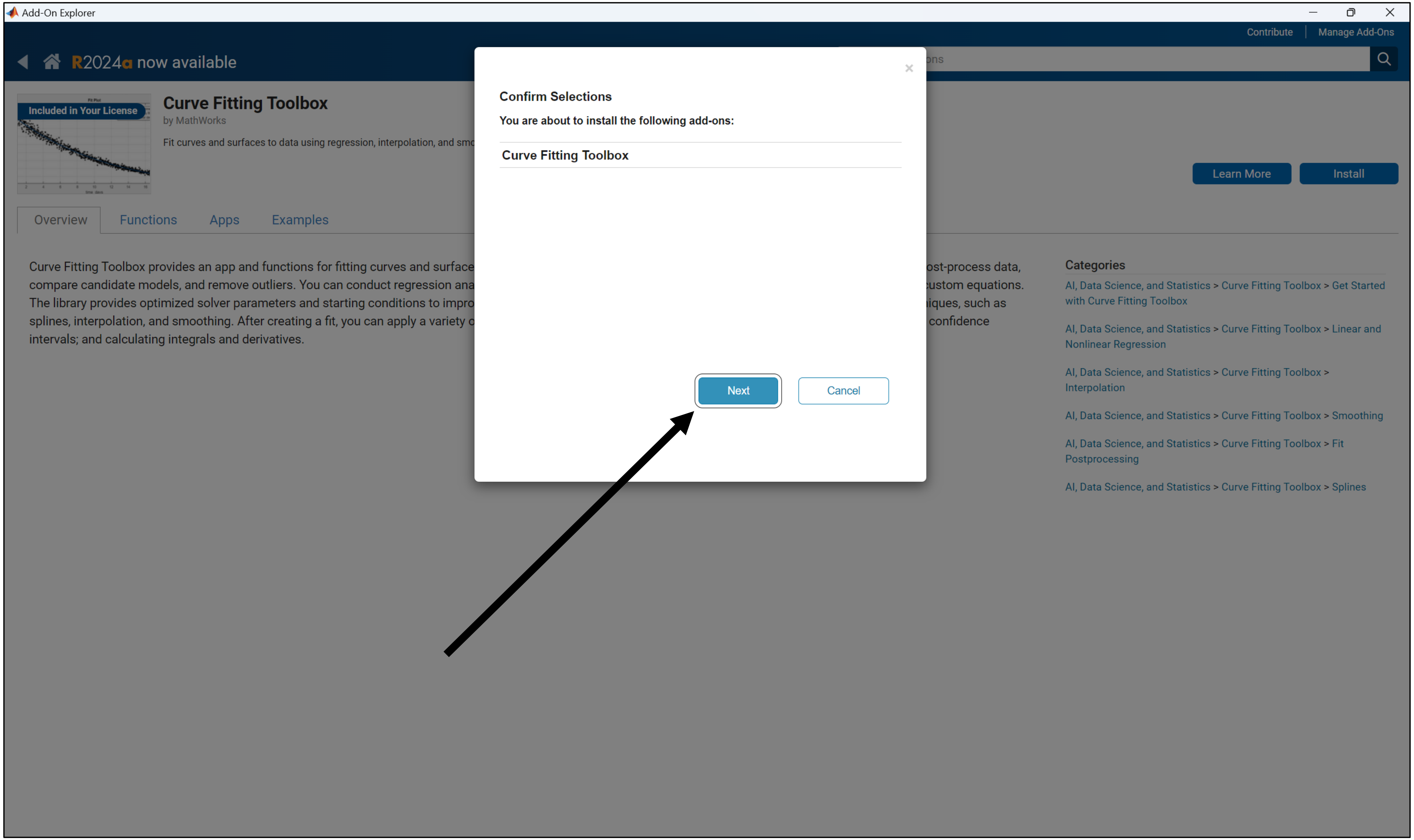

MATLAB R2023a - academic use
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Amanda
New Script
New Live Script
Open
Find Files
Compare
FILE
Import Data
Clean Data
Variable
Save Workspace
Clear Workspace
VARIABLE
Favorites
Analyze Code
Run and Time
Clear Commands
CODE
Simulink
SIMULINK
Layout
Preferences
Set Path
Add-Ons
ENVIRONMENT
Help
Community
Request Support
Learn MATLAB
RESOURCES
C: Users ablei Desktop SISMID Code forecasting_odemodels code
Current Folder
Name
plotFit_ODEModel.m
parameters - SEIURC.tif
parameterSearchODE.m
options_forecast_SEIR_flu1918_dist1_3.m
options_forecast_SEIR_flu1918_dist1_1.m
options_forecast_SEIR_flu1918_dist1_0.m
options_forecast_EXP_flu1918_dist1_3.m
options_fit_SWISS_COVID.m
options_fit_SEIR_unreported_covid_swiss_dist1_0.m
options_fit_SEIR_unreported_covid_spain_dist1_3.m
options_fit_SEIR_unreported_covid_spain_dist1_1.m
options_fit_SEIR_flu1918.m
options_fit_SEIR_flu1918_dist1_3.m
options_fit_SEIR_flu1918_dist1_1.m
options_fit_SEIR_flu1918_dist1_0.m
options_fit_SEIR_COVIDSwiss.m
options_fit_SEIR1.m
options_fit.m
initialParams.m
GLM.m
GGM.m
getMeanVarianceRatio.m
getAICc.m
GetAdjustedFontSize.m
get_nparams.m
Forecast 0-20.tif
fit_model.m
EXP.m
computeWIS.m
computeQuantiles.m
computeforecastperformance.m
composite1.m
AddErrorStructure.m
output
input
curve-flu1918SF.txt
curve-covid-firstwave-swiss-reporteddt.txt
GLM.m (Function)
GLM(t, x, params0)
Editor - C:\Users\ablei\Desktop\SISMID\Code\forecasting_odemodels code\GLM.m
SEIR_unreported_expo_decline.m
Running_Easy_Amanda.m
untitled7 *
options_forecast_SEIR_unreported_covid_swiss_dist1_0.m
GLM.m
% < Author: Gerardo Chowell
function dx=GLM(t,x,params0)
% params0(1)=r, params0(2)=p, params0(3)=K
% GLM model
dx(1,1)=params0(1).*(x(1,1).^params0(2)).*(1-(x(1,1)/params0(3)));
Command Window
composite12 =
7.3618 7.2891 7.3928
ans =
1.0e+03 *
0.0010 1.5468 1.5362 0.0106 0.0050
Work...
Name
ans

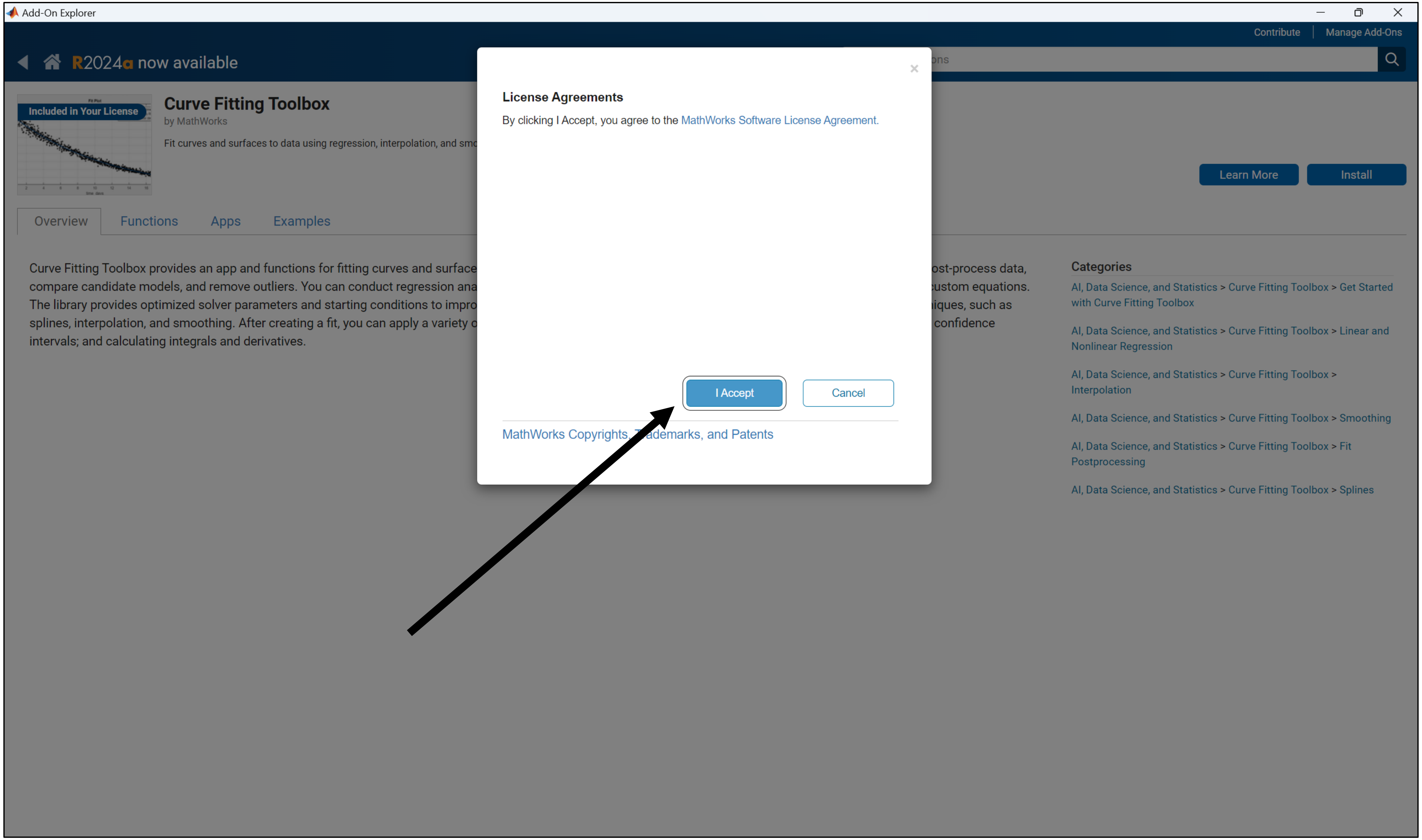

MATLAB R2023a - academic use
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Amanda
New Script
New Live Script
New
Open
Find Files
Compare
FILE
Import Data
Clean Data
Variable
Save Workspace
Clear Workspace
VARIABLE
Favorites
Analyze Code
Run and Time
Clear Commands
CODE
Simulink
SIMULINK
Layout
Preferences
Set Path
ENVIRONMENT
Add-Ons
Help
Community
Request Support
Learn MATLAB
Get Add-Ons
Manage Add-Ons
Package Toolbox
Package App
Get Hardware Support Packages
C: › Users › ablei › Desktop › SISMID › Code › forecasting_odemodels code
Current Folder
Editor - C:\Users\ablei\Desktop\SISMID\Code\forecasting_odemodels code\GLM.m
SEIR_unreported_expo_decline.m
Running_Easy_Amanda.m
options_forecast_SEIR_unreported_covid_swiss_dist1_0.m
GLM.m
% < Author: Gerardo Chowell
function dx=GLM(t,x,params0)
% params0(1)=r, params0(2)=p, params0(3)=K
% GLM model
dx(1,1)=params0(1).*(x(1,1).^params0(2)).*(1-(x(1,1)/params0(3)));
plotFit_ODEModel.m
parameters - SEIURC.tif
parameterSearchODE.m
options_forecast_SEIR_unreported_covid_swiss_dist1_0.m
options_forecast_SEIR_flu1918_dist1_3.m
options_forecast_SEIR_flu1918_dist1_1.m
options_forecast_SEIR_flu1918_dist1_0.m
options_forecast_EXP_flu1918_dist1_3.m
options_fit_SWISS_COVID.m
options_fit_SEIR_unreported_covid_swiss_dist1_0.m
options_fit_SEIR_unreported_covid_spain_dist1_3.m
options_fit_SEIR_unreported_covid_spain_dist1_1.m
options_fit_SEIR_flu1918.m
options_fit_SEIR_flu1918_dist1_3.m
options_fit_SEIR_flu1918_dist1_1.m
options_fit_SEIR_flu1918_dist1_0.m
options_fit_SEIR_COVIDSwiss.m
options_fit_SEIR1.m
options_fit.m
initialParams.m
GLM.m
GGM.m
getMeanVarianceRatio.m
getAICc.m
GetAdjustedFontSize.m
get_nparams.m
Forecast 0-20.tif
fit_model.m
EXP.m
computeWIS.m
computeQuantiles.m
computeforecastperformance.m
composite1.m
AddErrorStructure.m
output
input
curve-flu1918SF.txt
curve-covid-firstwave-swiss-reporteddt.txt
GLM.m (Function)
GLM(t, x, params0)
Command Window
composite12 =
7.3618 7.2891 7.3928
ans =
1.0e+03 *
0.0010 1.5468 1.5362 0.0106 0.0050
Work...
Name
ans

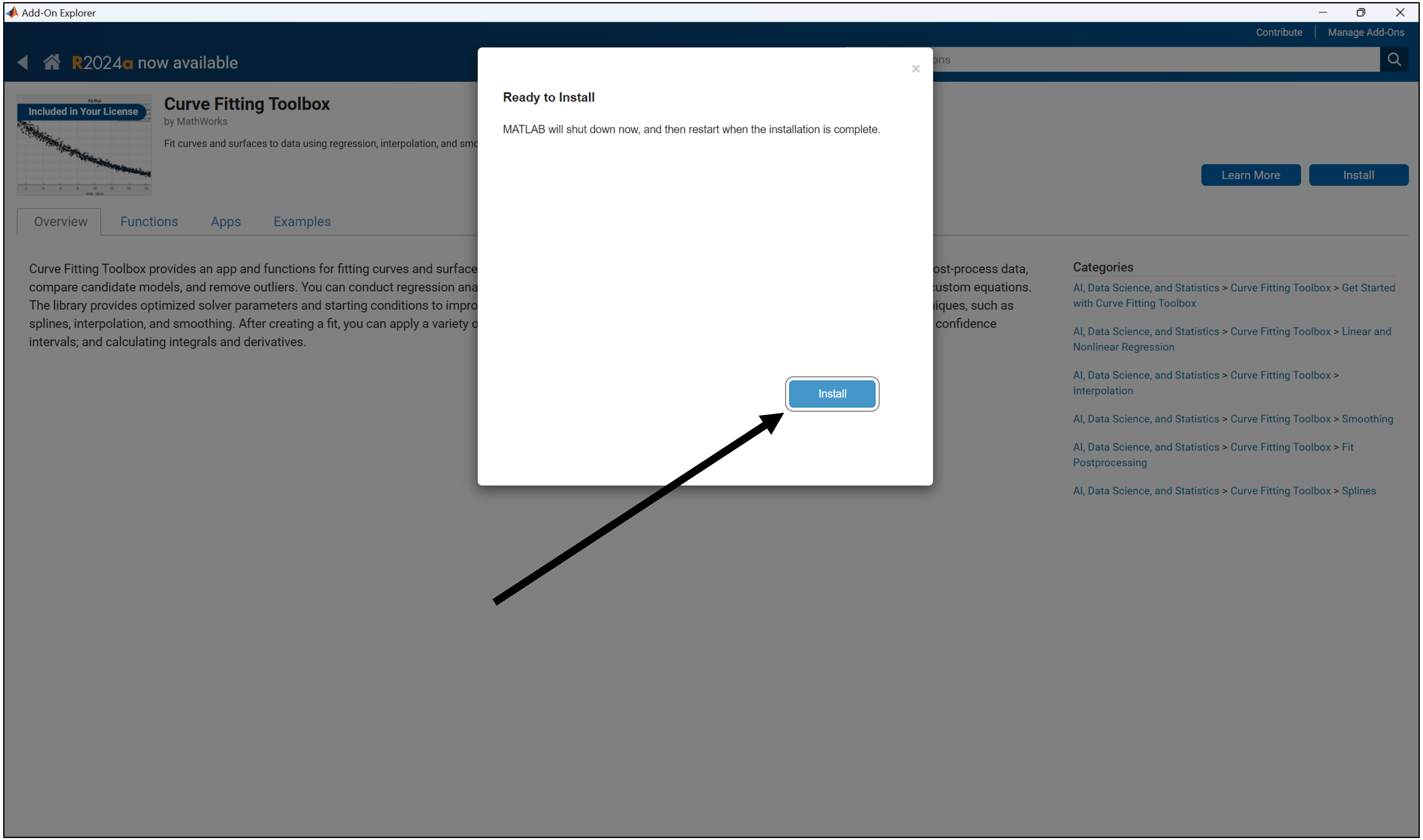

Add-On Explorer
Contribute
Manage Add-Ons
R2024a now available
Search for add-ons
For You
My Products 112
Recommendations 60
Filter by Source
MathWorks 403
Community 48,786
Filter by Category
Using MATLAB
MATLAB 11,694
Using Simulink
Simulink 844
Physical Modeling 2,216
Event-Based Modeling 68
Real-Time Simulation and Testing 35
Workflows
Parallel Computing 189
Reporting and Database Access 205
Systems Engineering 27
Code Generation 334
Application Deployment 124
Verification, Validation, and Test 172
Cloud Capabilities 70
Teaching and Learning 3
Applications
Explore Your Software
See all MathWorks products you can use.
View My Products
Recommended For You
Show All 60
Geoid Data for Aerospace Toolbox
Geoid Data for Aerospace Toolbox
Included in Your License
Audio Toolbox
Design and analyze speech, acoustic, and audio processing...
ECG simulation using MATLAB
This code generates all possible forms of ECG signals with the...
Google Earth Toolbox
Various plotting/drawing functions that can be saved as...
iPod Scramble Solution
Solves the Scramble game (like Boggle) by Zynga on my iPod.
MathWorks Toolboxes and Products
Show All 124
Included in Your License
Bioinformatics Toolbox
Included in Your License
Computer Vision Toolbox
Included in Your License
Deep Learning Toolbox
Included in Your License
GPU Coder


HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Amanda
New Script
New Live Script
New
Open
Find Files
Compare
Import
Clean
Variable
Save Workspace
Favorites
Analyze Code
Run and Time
Simulink
Layout
Preferences
Set Path
Add-Ons
Help
Community
Request Support
FILE
Current Folder
Name
plotFit_ODEModel.m
parameters - SEIURC.tif
parameterSearchODE.m
options_fit_SWISS_COVID.m
options_fit_SEIR_flu1918.m
options_fit_SEIR_COVIDSwiss.m
options_fit_SEIR1.m
options_fit.m
initialParams.m
GLM.m
GGM.m
getMeanVarianceRatio.m
getAICc.m
GetAdjustedFontSize.m
get_nparams.m
Forecast 0-20.tif
fit_model.m
EXP.m
computeWIS.m
computeQuantiles.m
computeforecastperformance.m
composite1.m
AddErrorStructure.m
output
input
curve-flu1918SF.txt
GLM.m (Function)
GLM(t, x, params0)
Add-On Explorer
Contribute
Manage Add-Ons
R2024a now available
curve fitting
For You
My Products 112
Recommendations 60
Filter by Source
MathWorks 403
Community 48,786
Filter by Category
Using MATLAB
MATLAB 11,694
Using Simulink
Simulink 844
Physical Modeling 2,216
Event-Based Modeling 68
Real-Time Simulation and Testing 35
Workflows
Parallel Computing 189
Reporting and Database Access 205
Systems Engineering 27
Code Generation 334
Application Deployment 124
Verification, Validation, and Test 172
Cloud Capabilities 70
Teaching and Learning 3
Applications
Explore Your Software
See all MathWorks products you can use.
Recommended For You
Geoid Data for Aerospace Toolbox
Geoid Data for Aerospace Toolbox
ECG simulation using MATLAB
This code generates all possible forms of ECG signals with the…
MathWorks Toolboxes and Products
Included in Your License
Bioinformatics Toolbox
Computer Vision Toolbox
Deep Learning Toolbox
GPU Coder
Add-Ons
Curve Fitting Toolbox
Product
EzyFit 2.44
A free curve fitting toolbox for Matlab
29k Downloads
gaussian curve fit
gaussian curve fit
34k Downloads
Multiple curve fitting with common parameters using NLINFIT
Wrapper for NLINFIT which allows simultaneous fitting for multiple dat…
3.9k Downloads
Evolutionary curve fitting
Particle swarm optimization is used to perform the thermal transient im…
1.7k Downloads
Suggestions
curve fittingsmoothen
fittingmethod
fittings
fittinganddesigningexperiments
fittingby
0.0010 1.5468 1.5362 0.0106 0.0050
Zoom: 80%
UTF-8
LF
GLM
Ln 15
Col 1
Work…
ans
GLM.m


Add-On Explorer
Contribute
Manage Add-Ons
R2024a now available
Search for add-ons
Included in Your License
Curve Fitting Toolbox
by MathWorks
Fit curves and surfaces to data using regression, interpolation, and smoothing
Learn More
Install
Overview
Functions
Apps
Examples
Curve Fitting Toolbox provides an app and functions for fitting curves and surfaces to data. The toolbox lets you perform exploratory data analysis, preprocess and post-process data, compare candidate models, and remove outliers. You can conduct regression analysis using the library of linear and nonlinear models provided or specify your own custom equations. The library provides optimized solver parameters and starting conditions to improve the quality of your fits. The toolbox also supports nonparametric modeling techniques, such as splines, interpolation, and smoothing. After creating a fit, you can apply a variety of post-processing methods for plotting, interpolation, and extrapolation; estimating confidence intervals; and calculating integrals and derivatives.
Categories
AI, Data Science, and Statistics > Curve Fitting Toolbox > Get Started with Curve Fitting Toolbox
AI, Data Science, and Statistics > Curve Fitting Toolbox > Linear and Nonlinear Regression
AI, Data Science, and Statistics > Curve Fitting Toolbox > Interpolation
AI, Data Science, and Statistics > Curve Fitting Toolbox > Smoothing
AI, Data Science, and Statistics > Curve Fitting Toolbox > Fit Postprocessing
AI, Data Science, and Statistics > Curve Fitting Toolbox > Splines


Add-On Explorer
Contribute
Manage Add-Ons
R2024a now available
Curve Fitting Toolbox
by MathWorks
Included in Your License
Fit curves and surfaces to data using regression, interpolation, and smo
Learn More
Install
Overview
Functions
Apps
Examples
Curve Fitting Toolbox provides an app and functions for fitting curves and surface
compare candidate models, and remove outliers. You can conduct regression ana
The library provides optimized solver parameters and starting conditions to impro
splines, interpolation, and smoothing. After creating a fit, you can apply a variety o
intervals; and calculating integrals and derivatives.
ost-process data,
custom equations.
iques, such as
confidence
Categories
AI, Data Science, and Statistics > Curve Fitting Toolbox > Get Started with Curve Fitting Toolbox
AI, Data Science, and Statistics > Curve Fitting Toolbox > Linear and Nonlinear Regression
AI, Data Science, and Statistics > Curve Fitting Toolbox > Interpolation
AI, Data Science, and Statistics > Curve Fitting Toolbox > Smoothing
AI, Data Science, and Statistics > Curve Fitting Toolbox > Fit Postprocessing
AI, Data Science, and Statistics > Curve Fitting Toolbox > Splines
Confirm Selections
You are about to install the following add-ons:
Curve Fitting Toolbox
Next
Cancel


Add-On Explorer
Contribute
Manage Add-Ons
R2024a now available
Included in Your License
Curve Fitting Toolbox
by MathWorks
Fit curves and surfaces to data using regression, interpolation, and smo
Learn More
Install
Overview
Functions
Apps
Examples
Curve Fitting Toolbox provides an app and functions for fitting curves and surface
compare candidate models, and remove outliers. You can conduct regression ana
The library provides optimized solver parameters and starting conditions to impro
splines, interpolation, and smoothing. After creating a fit, you can apply a variety o
intervals; and calculating integrals and derivatives.
ost-process data,
custom equations.
iques, such as
confidence
License Agreements
By clicking I Accept, you agree to the MathWorks Software License Agreement.
I Accept
Cancel
MathWorks Copyrights, Trademarks, and Patents
Categories
AI, Data Science, and Statistics > Curve Fitting Toolbox > Get Started with Curve Fitting Toolbox
AI, Data Science, and Statistics > Curve Fitting Toolbox > Linear and Nonlinear Regression
AI, Data Science, and Statistics > Curve Fitting Toolbox > Interpolation
AI, Data Science, and Statistics > Curve Fitting Toolbox > Smoothing
AI, Data Science, and Statistics > Curve Fitting Toolbox > Fit Postprocessing
AI, Data Science, and Statistics > Curve Fitting Toolbox > Splines


Add-On Explorer
Contribute
Manage Add-Ons
R2024a now available
Included in Your License
Curve Fitting Toolbox
by MathWorks
Fit curves and surfaces to data using regression, interpolation, and smo
Learn More
Install
Overview
Functions
Apps
Examples
Curve Fitting Toolbox provides an app and functions for fitting curves and surface
compare candidate models, and remove outliers. You can conduct regression ana
The library provides optimized solver parameters and starting conditions to impro
splines, interpolation, and smoothing. After creating a fit, you can apply a variety o
intervals; and calculating integrals and derivatives.
ost-process data,
custom equations.
iques, such as
confidence
Ready to Install
MATLAB will shut down now, and then restart when the installation is complete.
Install
Categories
AI, Data Science, and Statistics > Curve Fitting Toolbox > Get Started with Curve Fitting Toolbox
AI, Data Science, and Statistics > Curve Fitting Toolbox > Linear and Nonlinear Regression
AI, Data Science, and Statistics > Curve Fitting Toolbox > Interpolation
AI, Data Science, and Statistics > Curve Fitting Toolbox > Smoothing
AI, Data Science, and Statistics > Curve Fitting Toolbox > Fit Postprocessing
AI, Data Science, and Statistics > Curve Fitting Toolbox > Splines

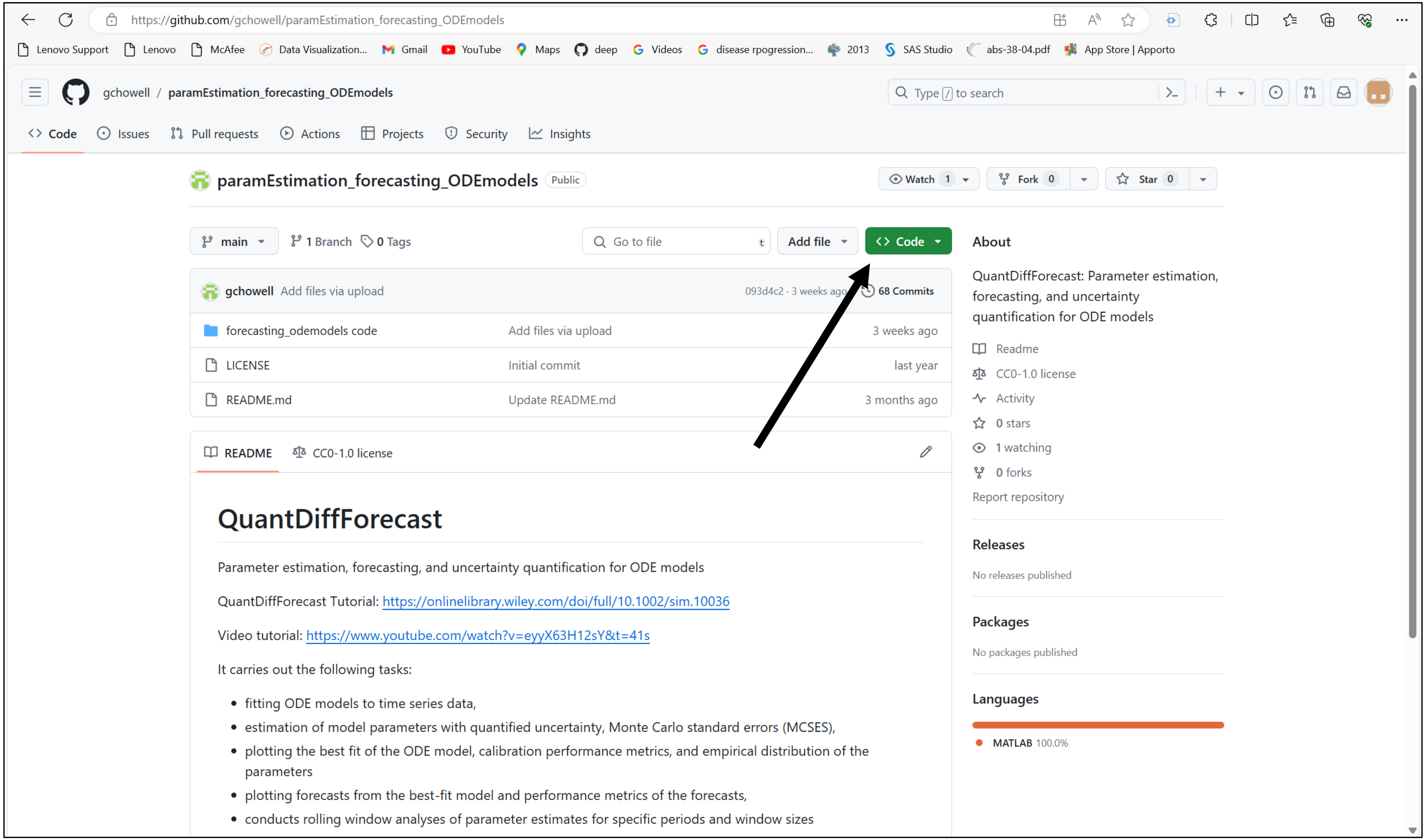
Please indicate if you have downloaded MATLAB and all necessary settings are selected/downloaded.

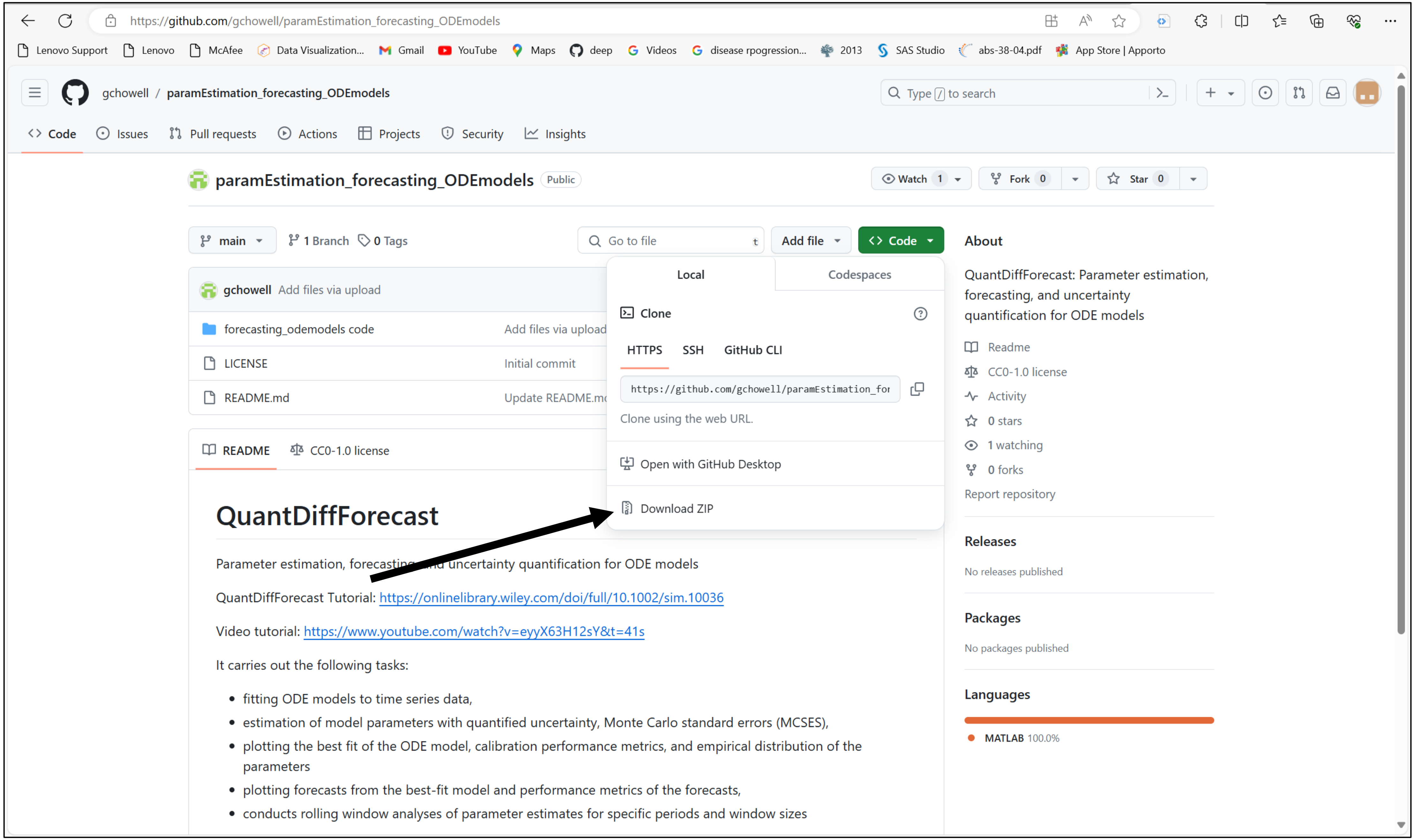
I am finished. — 0%

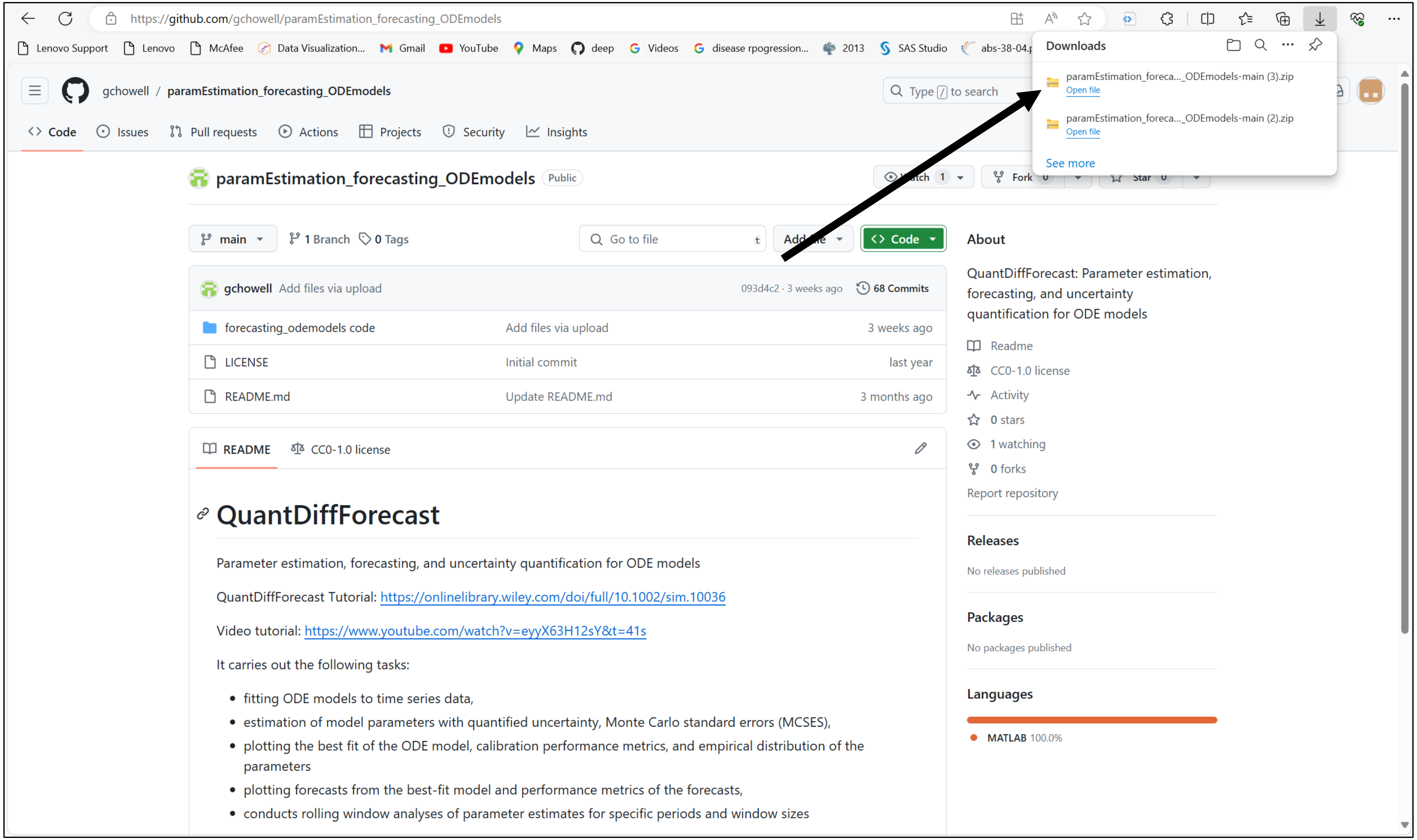
The *QuantDiffForecast* Toolbox

Multiple parameter estimation options	Flexible model specifications	Evaluation of bootstrapping error, model fit and forecasting performance
• Estimation: NLSQ & MLE • Error distributions: Normal, Poisson, and four variations of negative binomial • Bootstrapping and number of initial guesses	• User specified system of ODEs • Initial parameter and state variable conditions • Estimation of composite function(s) from the specified model parameters	• Mean Squared Error (MSE) • Mean Absolute Error (MAE) • Weighted Interval Scores (WIS) • 95% PI Coverage • Monte Carlo Squared Error (MCSE)

Downloading the Toolbox


https://github.com/gchowell/paramEstimation_forecasting_ODEmodels
Lenovo Support
Lenovo
McAfee
Data Visualization...
Gmail
YouTube
Maps
deep
Videos
disease rpogression...
2013
SAS Studio
abs-38-04.pdf
App Store | Apporto
gchowell / paramEstimation_forecasting_ODEmodels
Type / to search
Code
Issues
Pull requests
Actions
Projects
Security
Insights
paramEstimation_forecasting_ODEmodels
Public
Watch 1
Fork 0
Star 0
main
1 Branch
0 Tags
Go to file
t
Add file
Code
gchowell Add files via upload
093d4c2 · 3 weeks ago
68 Commits
forecasting_odemodels code
Add files via upload
3 weeks ago
LICENSE
Initial commit
last year
README.md
Update README.md
3 months ago
README
CC0-1.0 license
QuantDiffForecast
Parameter estimation, forecasting, and uncertainty quantification for ODE models
QuantDiffForecast Tutorial: https://onlinelibrary.wiley.com/doi/full/10.1002/sim.10036
Video tutorial: https://www.youtube.com/watch?v=eyyX63H12sY&t=41s
It carries out the following tasks:
fitting ODE models to time series data,
estimation of model parameters with quantified uncertainty, Monte Carlo standard errors (MCSES),
plotting the best fit of the ODE model, calibration performance metrics, and empirical distribution of the parameters
plotting forecasts from the best-fit model and performance metrics of the forecasts,
conducts rolling window analyses of parameter estimates for specific periods and window sizes
About
QuantDiffForecast: Parameter estimation, forecasting, and uncertainty quantification for ODE models
Readme
CC0-1.0 license
Activity
0 stars
1 watching
0 forks
Report repository
Releases
No releases published
Packages
No packages published
Languages
MATLAB 100.0%


https://github.com/gchowell/paramEstimation_forecasting_ODEmodels
Lenovo Support
Lenovo
McAfee
Data Visualization...
Gmail
YouTube
Maps
deep
Videos
disease rpogression...
2013
SAS Studio
abs-38-04.pdf
App Store | Apporto
gchowell / paramEstimation_forecasting_ODEmodels
Type / to search
Code
Issues
Pull requests
Actions
Projects
Security
Insights
paramEstimation_forecasting_ODEmodels
Public
Watch 1
Fork 0
Star 0
main
1 Branch
0 Tags
Go to file
t
Add file
Code
gchowell Add files via upload
forecasting_odemodels code
Add files via upload
LICENSE
Initial commit
README.md
Update README.md
Local
Codespaces
Clone
HTTPS
SSH
GitHub CLI
https://github.com/gchowell/paramEstimation_for
Clone using the web URL.
Open with GitHub Desktop
Download ZIP
README
CC0-1.0 license
QuantDiffForecast
Parameter estimation, forecasting and uncertainty quantification for ODE models
QuantDiffForecast Tutorial: https://onlinelibrary.wiley.com/doi/full/10.1002/sim.10036
Video tutorial: https://www.youtube.com/watch?v=eyyX63H12sY&t=41s
It carries out the following tasks:
fitting ODE models to time series data,
estimation of model parameters with quantified uncertainty, Monte Carlo standard errors (MCSES),
plotting the best fit of the ODE model, calibration performance metrics, and empirical distribution of the parameters
plotting forecasts from the best-fit model and performance metrics of the forecasts,
conducts rolling window analyses of parameter estimates for specific periods and window sizes
About
QuantDiffForecast: Parameter estimation, forecasting, and uncertainty quantification for ODE models
Readme
CC0-1.0 license
Activity
0 stars
1 watching
0 forks
Report repository
Releases
No releases published
Packages
No packages published
Languages
MATLAB 100.0%

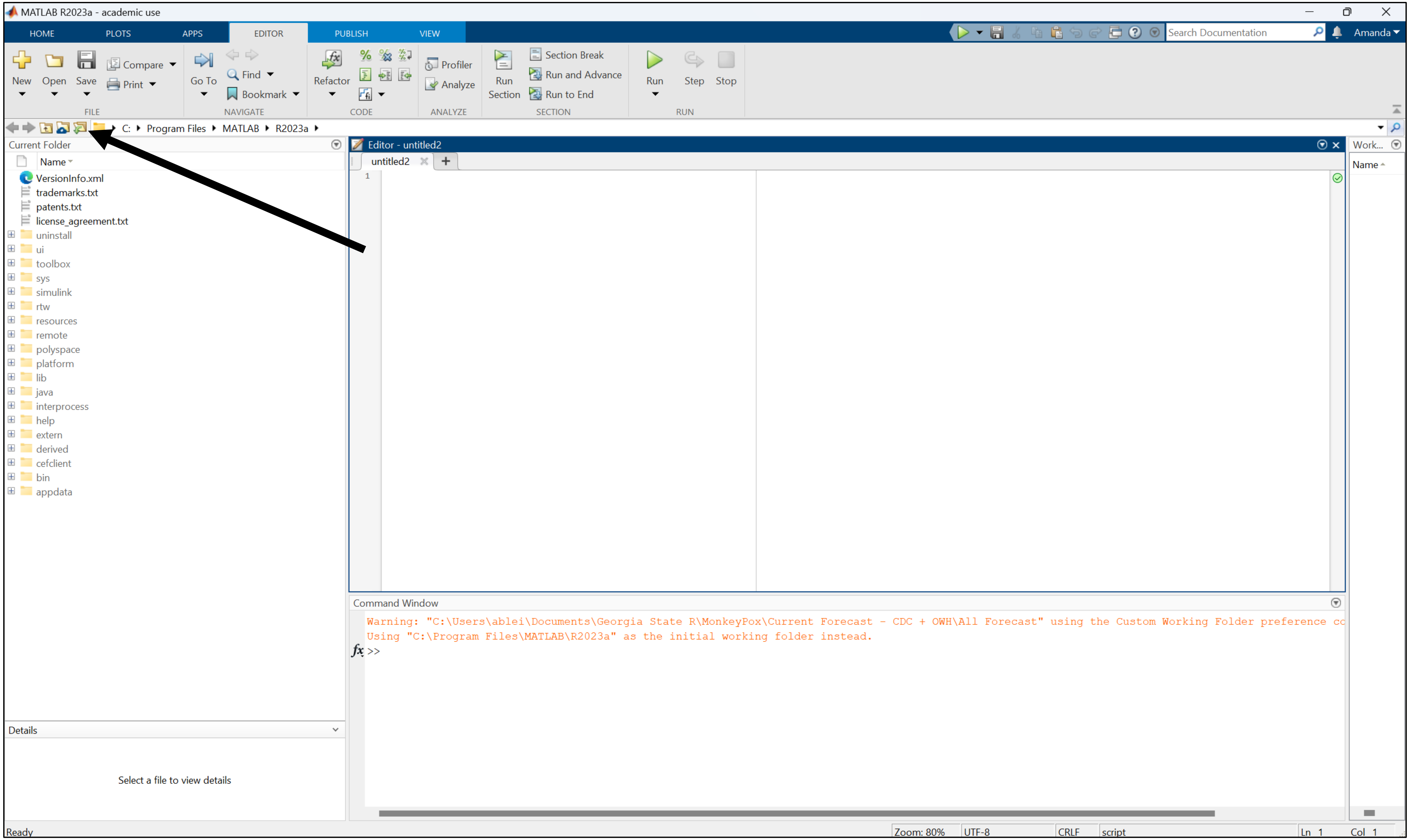

https://github.com/gchowell/paramEstimation_forecasting_ODEmodels
Lenovo Support
Lenovo
McAfee
Data Visualization...
Gmail
YouTube
Maps
deep
Videos
disease rpogression...
2013
SAS Studio
Downloads
paramEstimation_foreca..._ODEmodels-main (3).zip
Open file
paramEstimation_foreca..._ODEmodels-main (2).zip
Open file
See more
gchowell / paramEstimation_forecasting_ODEmodels
Type / to search
Code
Issues
Pull requests
Actions
Projects
Security
Insights
paramEstimation_forecasting_ODEmodels
Public
Fork 0
Star 0
main
1 Branch
0 Tags
Go to file
Code
gchowell Add files via upload
093d4c2 · 3 weeks ago
68 Commits
forecasting_odemodels code
Add files via upload
3 weeks ago
LICENSE
Initial commit
last year
README.md
Update README.md
3 months ago
README
CC0-1.0 license
QuantDiffForecast
Parameter estimation, forecasting, and uncertainty quantification for ODE models
QuantDiffForecast Tutorial: https://onlinelibrary.wiley.com/doi/full/10.1002/sim.10036
Video tutorial: https://www.youtube.com/watch?v=eyyX63H12sY&t=41s
It carries out the following tasks:
fitting ODE models to time series data,
estimation of model parameters with quantified uncertainty, Monte Carlo standard errors (MCSES),
plotting the best fit of the ODE model, calibration performance metrics, and empirical distribution of the parameters
plotting forecasts from the best-fit model and performance metrics of the forecasts,
conducts rolling window analyses of parameter estimates for specific periods and window sizes
About
QuantDiffForecast: Parameter estimation, forecasting, and uncertainty quantification for ODE models
Readme
CC0-1.0 license
Activity
0 stars
1 watching
0 forks
Report repository
Releases
No releases published
Packages
No packages published
Languages
MATLAB 100.0%

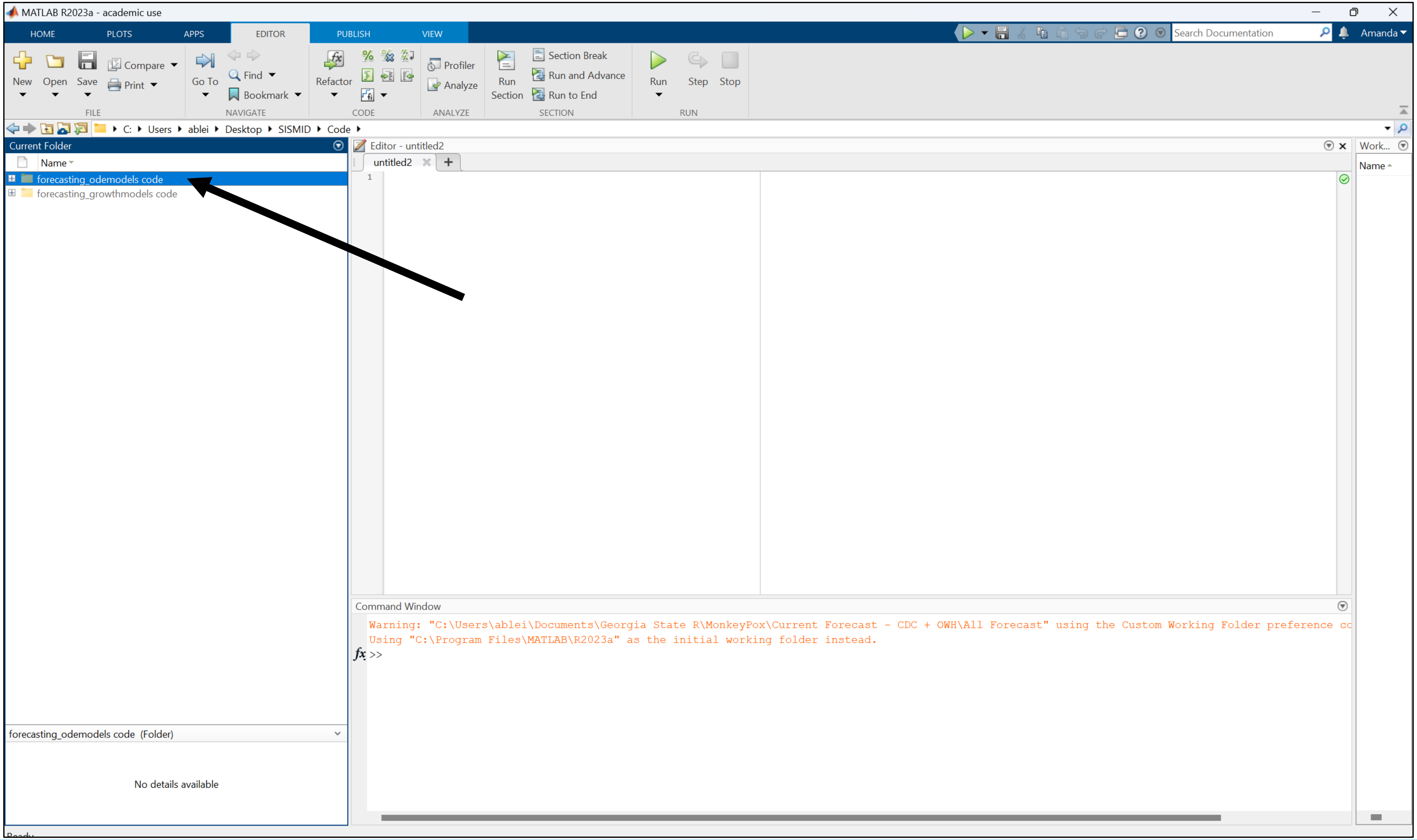
Name	Type	Compressed size	Password pr...	Size	Ratio	Date modified
paramEstimation_forecasting_ODEmodels-main	File folder					6/6/2024 9:52 AM

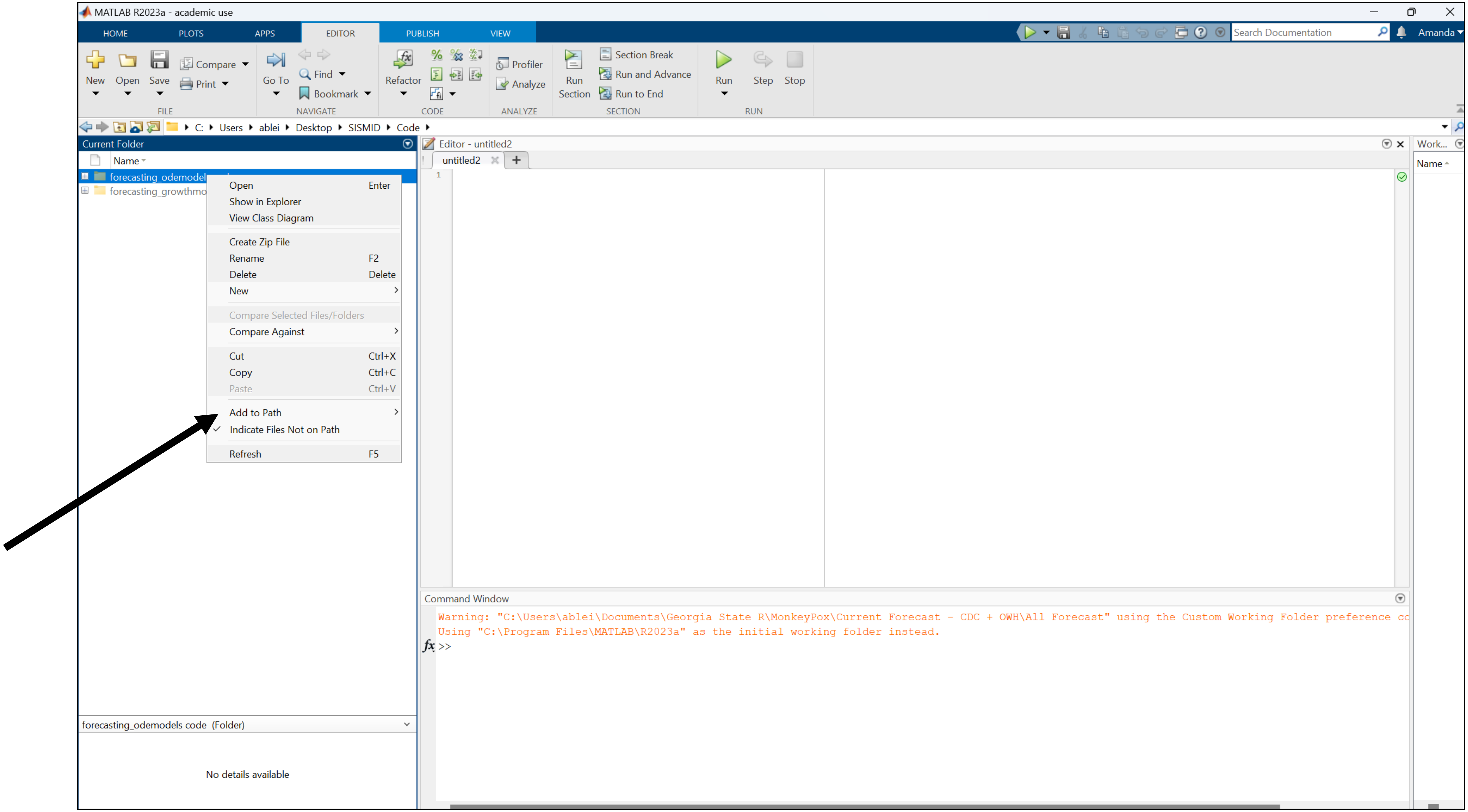
Name	Type	Compressed size	Password pr...	Size	Ratio	Date modified
forecasting_odemodels code	File folder					6/6/2024 9:52 AM
LICENSE	File	3 KB	No	7 KB	61%	6/6/2024 9:52 AM
README.md	MD File	1 KB	No	2 KB	50%	6/6/2024 9:52 AM

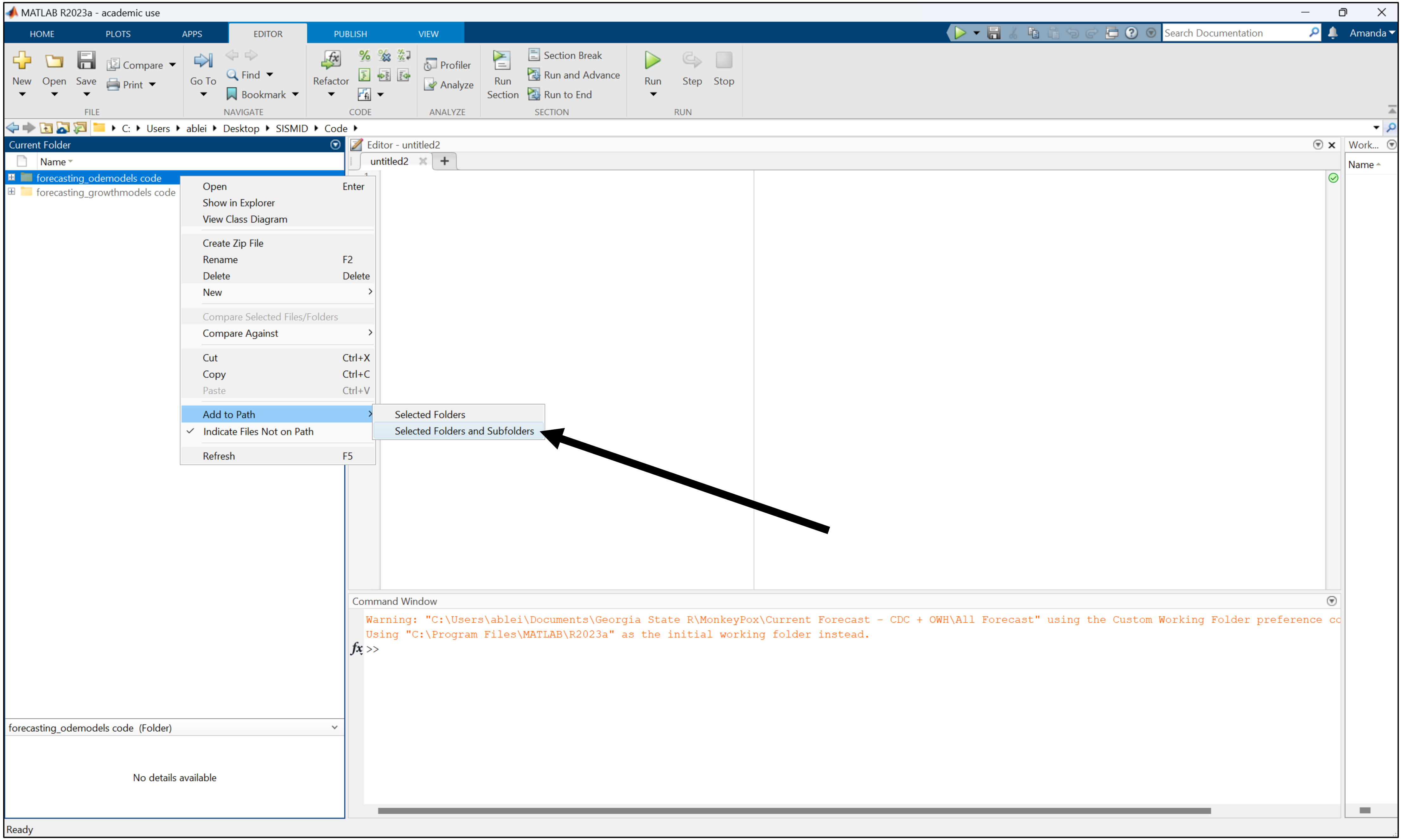
Name	Type	Compressed size	Password pr...	Size	Ratio	Date modified
input	File folder					6/6/2024 9:52 AM
AddErrorStructure	MATLAB Code	1 KB	No	4 KB	78%	6/6/2024 9:52 AM
composite1	MATLAB Code	1 KB	No	1 KB	38%	6/6/2024 9:52 AM
computeforecastperformance	MATLAB Code	1 KB	No	4 KB	68%	6/6/2024 9:52 AM
computeQuantiles	MATLAB Code	1 KB	No	2 KB	60%	6/6/2024 9:52 AM
computeWIS	MATLAB Code	2 KB	No	4 KB	71%	6/6/2024 9:52 AM
EXP	MATLAB Code	1 KB	No	1 KB	65%	6/6/2024 9:52 AM
fit_model	MATLAB Code	3 KB	No	8 KB	64%	6/6/2024 9:52 AM
get_nparams	MATLAB Code	1 KB	No	1 KB	58%	6/6/2024 9:52 AM
GetAdjustedFontSize	MATLAB Code	1 KB	No	1 KB	47%	6/6/2024 9:52 AM
getAICc	MATLAB Code	1 KB	No	2 KB	68%	6/6/2024 9:52 AM
getMeanVarianceRatio	MATLAB Code	1 KB	No	1 KB	74%	6/6/2024 9:52 AM
GGM	MATLAB Code	1 KB	No	1 KB	64%	6/6/2024 9:52 AM
GLM	MATLAB Code	1 KB	No	1 KB	66%	6/6/2024 9:52 AM
initialParams	MATLAB Code	1 KB	No	1 KB	71%	6/6/2024 9:52 AM
options_fit	MATLAB Code	2 KB	No	6 KB	71%	6/6/2024 9:52 AM
options_fit_SEIR_flu1918	MATLAB Code	2 KB	No	6 KB	70%	6/6/2024 9:52 AM
options_fit_SEIR_flu1918_dist1_0	MATLAB Code	2 KB	No	6 KB	70%	6/6/2024 9:52 AM
options_fit_SEIR_flu1918_dist1_1	MATLAB Code	2 KB	No	5 KB	70%	6/6/2024 9:52 AM
options_fit_SEIR_flu1918_dist1_3	MATLAB Code	2 KB	No	5 KB	70%	6/6/2024 9:52 AM
options_fit_SEIR_unreported_covid_sp...	MATLAB Code	2 KB	No	6 KB	69%	6/6/2024 9:52 AM
options_fit_SEIR_unreported_covid_sp...	MATLAB Code	2 KB	No	6 KB	69%	6/6/2024 9:52 AM
options_fit_SEIR1	MATLAB Code	2 KB	No	6 KB	70%	6/6/2024 9:52 AM
options_forecast_EXP_flu1918_dist1_3	MATLAB Code	2 KB	No	6 KB	71%	6/6/2024 9:52 AM
options_forecast_SEIR_flu1918_dist1_0	MATLAB Code	2 KB	No	6 KB	71%	6/6/2024 9:52 AM
options_forecast_SEIR_flu1918_dist1_3	MATLAB Code	2 KB	No	6 KB	70%	6/6/2024 9:52 AM

Setting up the Working Directory


MATLAB R2023a - academic use
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Amanda
New
Open
Save
Compare
Print
FILE
Go To
Find
Bookmark
NAVIGATE
Refactor
CODE
Profiler
Analyze
ANALYZE
Run Section
Section Break
Run and Advance
Run to End
SECTION
Run
Step
Stop
RUN
C: ▸ Program Files ▸ MATLAB ▸ R2023a ▸
Current Folder
Name
VersionInfo.xml
trademarks.txt
patents.txt
license_agreement.txt
uninstall
ui
toolbox
sys
simulink
rtw
resources
remote
polyspace
platform
lib
java
interprocess
help
extern
derived
cefclient
bin
appdata
Details
Select a file to view details
Editor - untitled2
untitled2
Work...
Name
Command Window
Warning: "C:\Users\ablei\Documents\Georgia State R\MonkeyPox\Current Forecast - CDC + OWH\All Forecast" using the Custom Working Folder preference cc
Using "C:\Program Files\MATLAB\R2023a" as the initial working folder instead.
>>
Ready
Zoom: 80%
UTF-8
CRLF
script
Ln 1
Col 1


MATLAB R2023a - academic use
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Amanda
New
Open
Save
Compare
Print
FILE
Go To
Find
Bookmark
NAVIGATE
Refactor
CODE
Profiler
Analyze
ANALYZE
Run Section
Section Break
Run and Advance
Run to End
SECTION
Run
Step
Stop
RUN
C: ▸ Users ▸ ablei ▸ Desktop ▸ SISMID ▸ Code ▸
Current Folder
Name
forecasting_odemodels code
forecasting_growthmodels code
forecasting_odemodels code (Folder)
No details available
Editor - untitled2
untitled2
Work...
Name
Command Window
Warning: "C:\Users\ablei\Documents\Georgia State R\MonkeyPox\Current Forecast - CDC + OWH\All Forecast" using the Custom Working Folder preference cc
Using "C:\Program Files\MATLAB\R2023a" as the initial working folder instead.
fx >>


MATLAB R2023a - academic use
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Amanda
New
Open
Save
Compare
Print
FILE
Go To
Find
Bookmark
NAVIGATE
Refactor
CODE
Profiler
Analyze
ANALYZE
Run Section
Section Break
Run and Advance
Run to End
SECTION
Run
Step
Stop
RUN
C: ▸ Users ▸ ablei ▸ Desktop ▸ SISMID ▸ Code ▸
Current Folder
Name
forecasting_odemodel
forecasting_growthmo
Open Enter
Show in Explorer
View Class Diagram
Create Zip File
Rename F2
Delete Delete
New
Compare Selected Files/Folders
Compare Against
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Add to Path
Indicate Files Not on Path
Refresh F5
Editor - untitled2
untitled2
Work...
Name
Command Window
Warning: "C:\Users\ablei\Documents\Georgia State R\MonkeyPox\Current Forecast - CDC + OWH\All Forecast" using the Custom Working Folder preference cc
Using "C:\Program Files\MATLAB\R2023a" as the initial working folder instead.
fx >>
forecasting_odemodels code (Folder)
No details available


MATLAB R2023a - academic use
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Amanda
New
Open
Save
Compare
Print
FILE
Go To
Find
Bookmark
NAVIGATE
Refactor
CODE
Profiler
Analyze
ANALYZE
Run Section
Section Break
Run and Advance
Run to End
SECTION
Run
Step
Stop
RUN
C: ▸ Users ▸ ablei ▸ Desktop ▸ SISMID ▸ Code ▸
Current Folder
Name
forecasting_odemodels code
forecasting_growthmodels code
Open Enter
Show in Explorer
View Class Diagram
Create Zip File
Rename F2
Delete Delete
New
Compare Selected Files/Folders
Compare Against
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Add to Path
Indicate Files Not on Path
Refresh F5
Selected Folders
Selected Folders and Subfolders
Editor - untitled2
untitled2
Work...
Name
Command Window
Warning: "C:\Users\ablei\Documents\Georgia State R\MonkeyPox\Current Forecast - CDC + OWH\All Forecast" using the Custom Working Folder preference cc
Using "C:\Program Files\MATLAB\R2023a" as the initial working folder instead.
fx >>
forecasting_odemodels code (Folder)
No details available
Ready

Please indicate if you have downloaded the toolbox, and have set-up your working directory.

Yes, I am done.

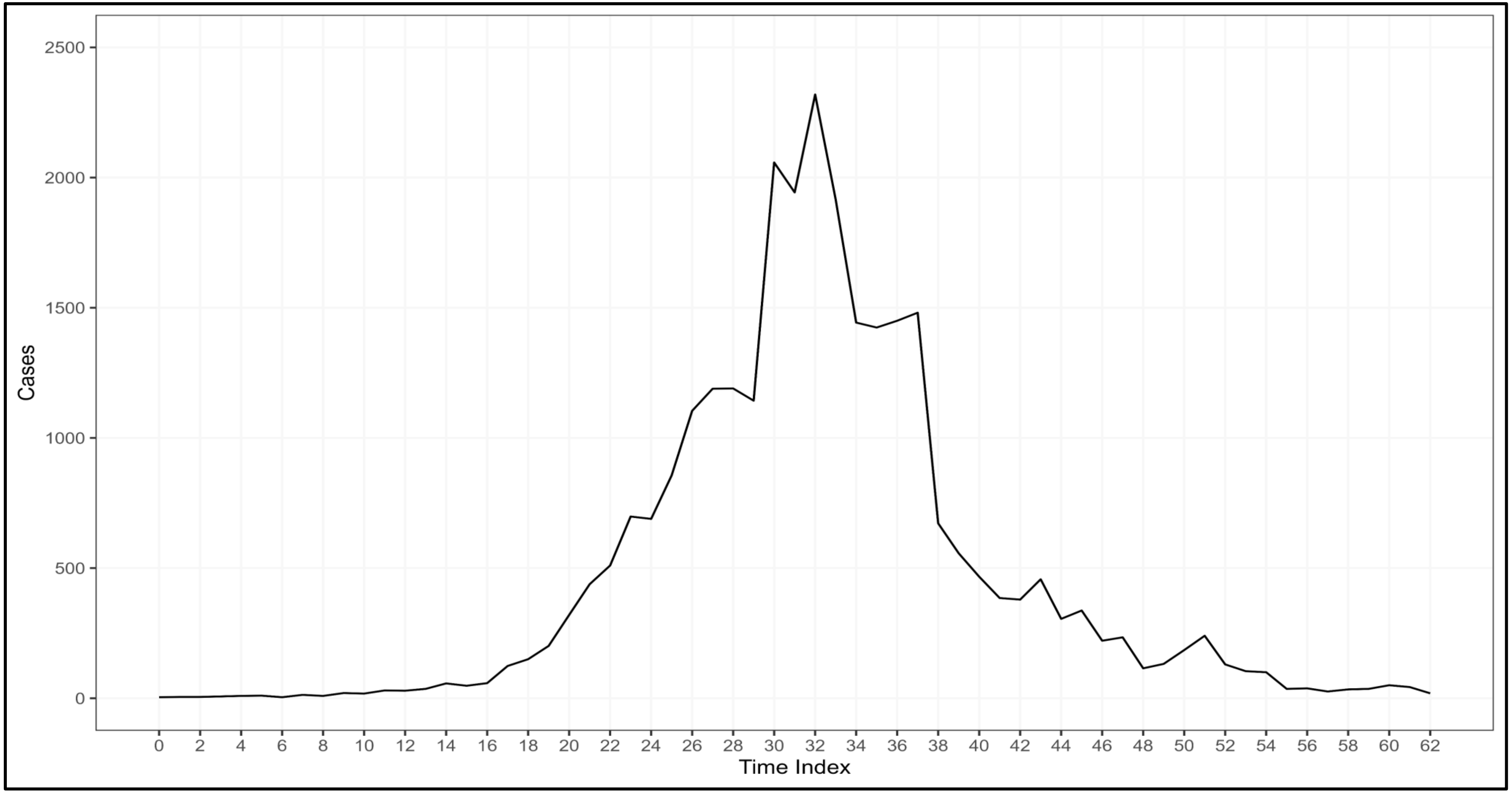
0%

Preparing for the Tutorial

Workflow

(1) Specifying the ODE Model and Preparing the Data → (2) Model Code and Testing → (3) Fitting the model to data with quantified uncertainty → (4) Plot the mean model fits and compute calibration period performance metrics → (5) Generating and evaluating model-based forecasts → (6) Plotting the model-based forecasts and performance metrics

Tutorial #1: 1918 Influenza



Daily incident curve of the fall wave of the 1918 influenza pandemic in San Francisco.

Data obtained from Chowell et al. JR Society Interface. 2007

Preparing the data

Set-Up

- Two folders are needed in your working directory: (1) input & (2) output

- Data goes in the input folder

File Name

- No formal file naming conventions must be followed IF working with incident data

- If working with CUMULATIVE incidence count data, the name of the time series data file must start with cumulative

File Structure

- 1st Column: Time Index (0, 1, 2…)

- 2nd Column: Temporal Incidence

- Columns should NOT have headers

- Data should be in *.txt* format

curve-flu1918SF

	A	B
1	0	4
2	1	5
3	2	5
4	3	7
5	4	9
6	5	10
7	6	4
8	7	13
9	8	9
10	9	20
11	10	18
12	11	30
13	12	29
14	13	36
15	14	57
16	15	48
17	16	58
18	17	124
19	18	150
20	19	201
21	20	320
22	21	438
23	22	510
24	23	698
25	24	689
26	25	856
27	26	1104
28	27	1189
29	28	1190
30	29	1143
31	30	2058
32	31	1943
33	32	2319
34	33	1916
35	34	1443
36	35	1424
37	36	1450

Specifying the ODE Model

Specifying the SEIR Model

```
% <==========================================================================
% < Author: Gerardo Chowell  ========================================
% <==========================================================================

function dx=SEIR1(t,x,params0,extra0)

%  params0(1) = beta, params0(2)=k,  params0(3)=gamma, params0(4)=N

dx=zeros(5,1);  % define the vector of the state derivatives

dx(1,1)= -params0(1)*x(1,1).*x(3,1)./params0(4); %S
dx(2,1)= params0(1)*x(1,1).*x(3,1)/params0(4) -params0(2)*x(2,1); %E
dx(3,1)= params0(2)*x(2,1) - params0(3)*x(3,1); %I
dx(4,1)= params0(3)*x(3,1); %R
dx(5,1)= params0(2)*x(2,1); %cumulative infections
```

$$
\begin{cases}
\dot{S} = -\beta S(t)\dfrac{I(t)}{N} \\
\dot{E} = \beta S(t)\dfrac{I(t)}{N} - \kappa E(t) \\
\dot{I} = \kappa E(t) - \gamma I(t) \\
\dot{R} = \gamma I(t) \\
\dot{C} = \kappa E(t)
\end{cases}
$$

Specifying the R0 Function

- In addition to specifying the ODE model, users can choose to specify a composite function using parameters specified in the ODE file

- For this tutorial, we use the following calculation for R0:

```
function composite_val=R0s(params0)

% beta(1), k(2), gamma(3), N(4)
composite_val=params0(:,1)./params0(:,3);
```

$$R0 = \frac{\beta}{\gamma}$$

Preparing the `options_fit.m` file

Overview

- Prior to fitting the model to data, multiple parameters must be specified in an `options_fit.m` structured file.
  - Name of `options_fit` file can be customized.
- Required Specification Sections :
  - Dataset Properties
  - Parameter Estimation
  - ODE Model
  - Rolling Window Analysis

Tutorial: We will be using two different `options_fit.m` files for this tutorial:

(1) `options_fit_SEIR_flu1918_dist1_1`
(2) `options_fit_SEIR_flu1918_dist1_3`

```
% <==========================================================================>
% <=================== Declare global variables =================================>
% <==========================================================================>

global method1 % Parameter estimation method

% <==========================================================================>
% <=============================== Datasets properties ======================>
% <==========================================================================>
% Located in the input folder, the time series data file is a text file with extension *.txt.
% The time series data file contains the incidence curve of the epidemic of interest.
% The first column corresponds to time index: 0,1,2, ... and the second
% column corresponds to the observed time series data.

cadfilename1='curve-flu1918SF'

caddisease='1918 Flu'; % string indicating the name of the disease related to the time series data

datatype='cases'; % string indicating the nature of the data (cases, deaths, hospitalizations, etc)
```

(1) `cadfilename1`: The name of the data file.

(2) `caddisease`: Name of the process of interest

(3) `datatype`: Nature of the data

```matlab
% <===========================================================================>
% <========================== Parameter estimation ===========================>
% <===========================================================================>

method1=1; % Type of estimation method

% Nonlinear least squares (LSQ)=0,
% MLE Poisson=1,
% MLE (Neg Binomial)=3, with VAR=mean+alpha*mean;
% MLE (Neg Binomial)=4, with VAR=mean+alpha*mean^2;
% MLE (Neg Binomial)=5, with VAR=mean+alpha*mean^d;


dist1=1; % Define dist1 which is the type of error structure. See below:

%dist1=0; % Normal distribution to model error structure (method1=0)
%dist1=1; % Poisson error structure (method1=0 OR method1=1)
%dist1=2; % Neg. binomial error structure where var = factor1*mean where
                  % factor1 is empirically estimated from the time series
                  % data (method1=0)
%dist1=3; % MLE (Neg Binomial) with VAR=mean+alpha*mean  (method1=3)
%dist1=4; % MLE (Neg Binomial) with VAR=mean+alpha*mean^2 (method1=4)
%dist1=5; % MLE (Neg Binomial)with VAR=mean+alpha*mean^d (method1=5)


switch method1
    case 1
        dist1=1;
    case 3
        dist1=3;
    case 4
        dist1=4;
    case 5
        dist1=5;
end


numstartpoints=20; % Number of initial guesses for optimization procedure using MultiStart

B=300; % number of bootstrap realizations to characterize parameter uncertainty
```

(1) `method1`: The type of estimation method.

- <u>Five options available:</u> NLSQ, MLE Poisson, and three MLE Neg. Binomial options
- The estimation method option should match the appropriate error structure.

(2) `dist1`: Error structure

- Six options available: Normal, Poisson, and four Neg. Binomial options

(3) `numstartpoints`: Number of initial guesses for optimization procedure

(4) `B`: Number of bootstrap realizations

*<u>Tutorial:</u> As we are working with smaller case counts, we will stick with Poisson and Negative Binomial error structures.

Specifying the ODE Model – `options_fit.m`

```
% <==============================================================================>
% <============================== ODE model =====================================>
% <==============================================================================>

model.fc=@SEIR1; % name of the model function
model.name='SEIR model';   % string indicating the name of the ODE model

params.label={'\beta','\kappa','\gamma','N'};  % list of symbols to refer to the model parameters
params.LB=[0.01 0.01 0.01 20]; % lower bound values of the parameter estimates
params.UB=[10 2 2 1000000]; % upper bound values of the parameter estimates
params.initial=[0.6 1/1.9 1/4.1 550000]; % initial parameter values/guesses
params.fixed=[0 1 1 1]; % Boolean vector to indicate any parameters that should remain fixed (1) to initial values indicated in params.initial. Otherwise the parameter i
params.fixI0=1; % Boolean variable indicating if the initial value of the fitting variable is fixed according to the first observation in the time series (1). Otherwise,
params.composite=@R0s;  % Estimate a composite function of the individual model parameter estimates otherwise it is left empty.
params.composite_name='R_0'; % Name of the composite parameter
params.extra0=[];

vars.label={'S','E','I','R','C'}; % list of symbols to refer to the variables included in the model
vars.initial=[params.initial(4)-4 0 4 0 4];  % vector of initial conditions for the model variables
vars.fit_index=5; % index of the model's variable that will be fit to the observed time series data
vars.fit_diff=1; % boolean variable to indicate if the derivative of model's fitting variable should be fit to data.
```

```
% <==================================================================================>
% <========================== Parameters of the rolling window analysis =========================>
% <==================================================================================>

windowsize1=17;  % moving window size

tstart1=1; % time point for the start of rolling window analysis

tend1=1;  %time point for the end of the rolling window analysis

printscreen1=1;
```

(1) `windowsize1`: The size of the moving window or calibration period

(2) `tstart1`: Start of the rolling window analysis

(3) `tend1`: End of the rolling window analysis

(4) `printscreen1`: Show the plots at the conclusion of running code

Rolling Window Analysis

- A rolling window analysis can be useful to assess the stability of the model parameters and forecasts over
  - Controls the data used to calibration the model (i.e., data that goes into the model)

Example: `tstart1 = 1, tend1 = 2, windowsize1 = 5`

Time	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Cases	1	2	5	8	9	1	2	1	1	5	3	6	9	4	8	0

Rolling Window #1

Rolling Window #2

Given tstart1 = 3, tend1 = 3, and windowsize1 = 5, what is the start and end time index of the rolling window analysis?

0

(A) 3, 3 — 0%

(B) 1, 2 — 0%

(C) 1, 3 — 0%

(D) 4, 5 — 0%

(E) I'm not sure. — 0%

Time	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Cases	1	2	5	8	9	1	2	1	1	5	3	6	9	4	8	0

Generating preliminary model solutions

plotODEModel()

Overview

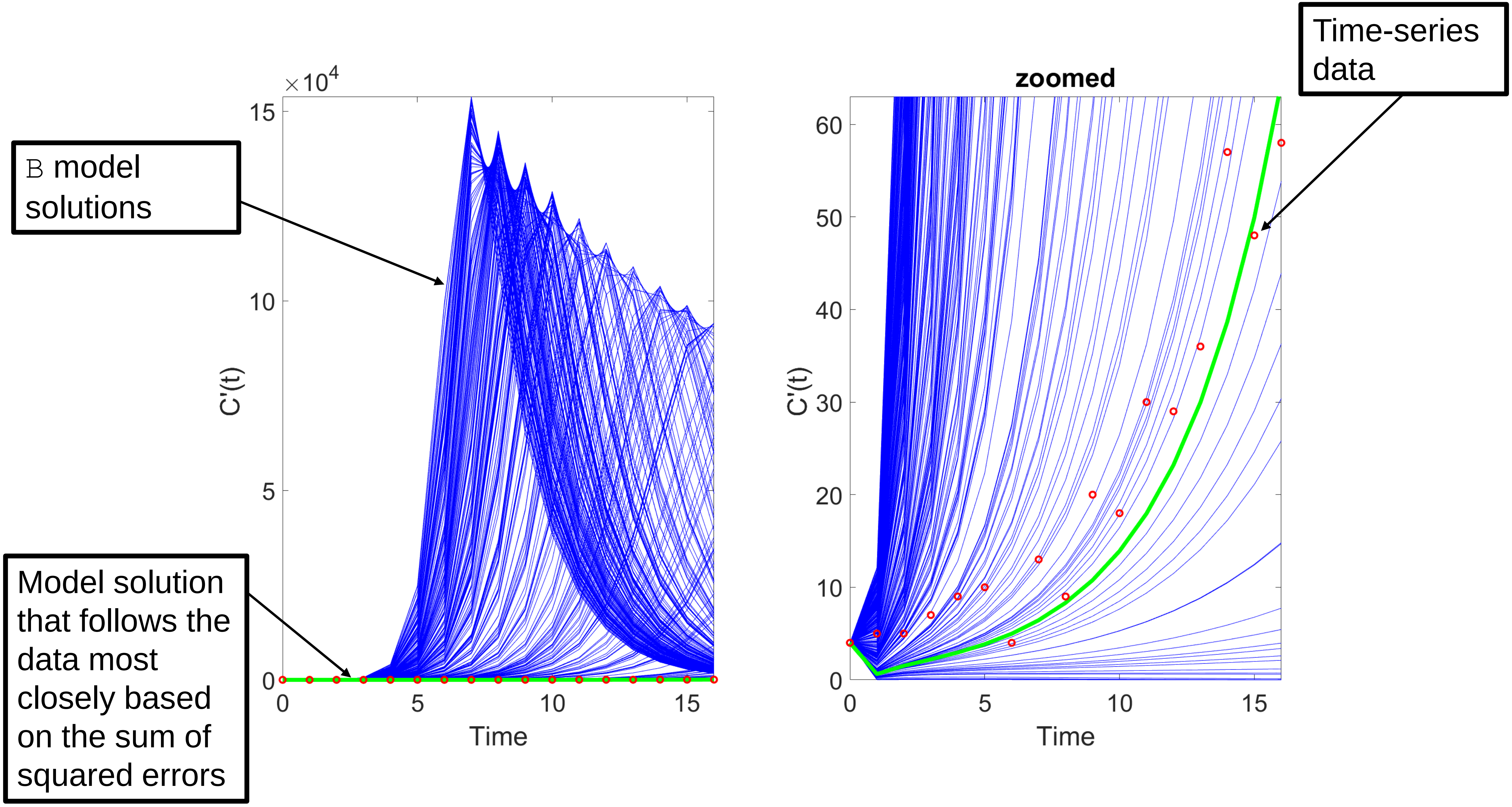
- Its often helpful to check that the user has correctly specified the model by checking that the model's solutions for the parameter ranges specified by the user correspond to a broad range of expected solutions

```
plotODEModel(@optionsFitFileName)
```

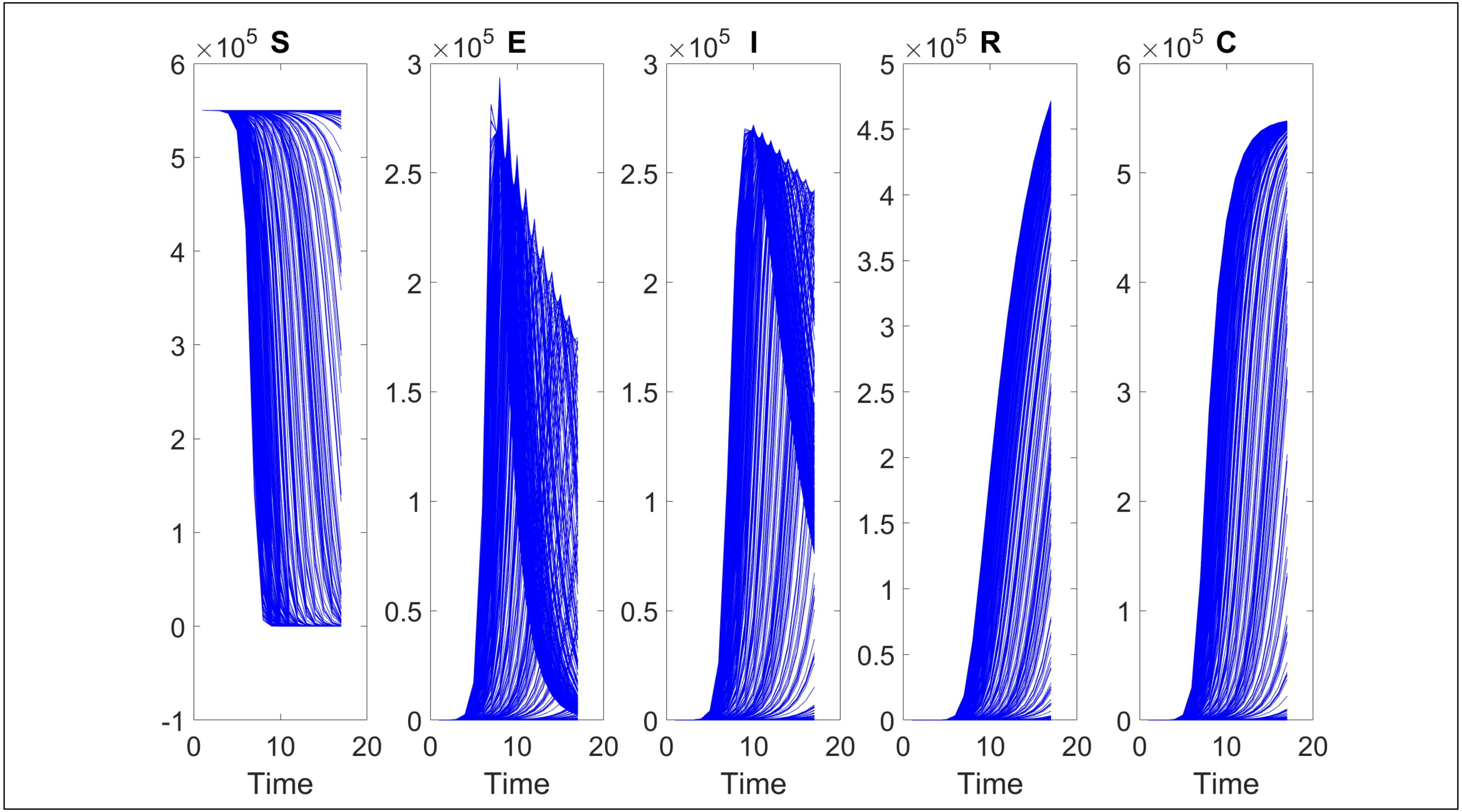
- Produces plots showing model solutions of state variables by generating a random sample of parameter sets from the specified parameter ranges and time-period

Tutorial Code Call

```
plotODEModel(@options_fit_SEIR_flu1918_dist1_1)
```



𝔹 model solutions
Model solution that follows the data most closely based on the sum of squared errors
×10^4
15
10
5
0
C'(t)
Time
zoomed
Time-series data
60
50
40
30
20
10
0
5
10
15



S
E
I
R
C
Time
Time
Time
Time
Time

Fitting, plotting, and evaluating the model with quantified uncertainty

Poisson and Neg. Binomial Error Structures

Step One: Fitting the Model to Data

- Prior to obtaining model fit files and plotting the model fit, we first must fit the model to data with quantified uncertainty using the following code call:

```
Run_Fit_ODEModel(@OptionsFitFileName,tstart1,tend1,windowsize1)
```

- `tstart1`, `tend1`, and `windowsize1` correspond to the values entered in the rolling window analysis section of the `options_fit.m` file.

Tutorial Code: (1) Poisson & (2) Negative Binomial

```
(1)Run_Fit_ODEModel(@options_fit_SEIR_flu1918_dist1_1, 1, 1, 17)
(2)Run_Fit_ODEModel(@options_fit_SEIR_flu1918_dist1_3, 1, 1, 17)
```

Step Two: Plotting the model fit

- After obtaining the model fits, we can obtain information related to: (1) model fit, (2) model parameter estimates, (3) Monte Carlo standard errors, (4) AICc values, (5) calibration performance metrics, & (6) composite parameter information for the specified model and data using the following call (`.csv` files in output)

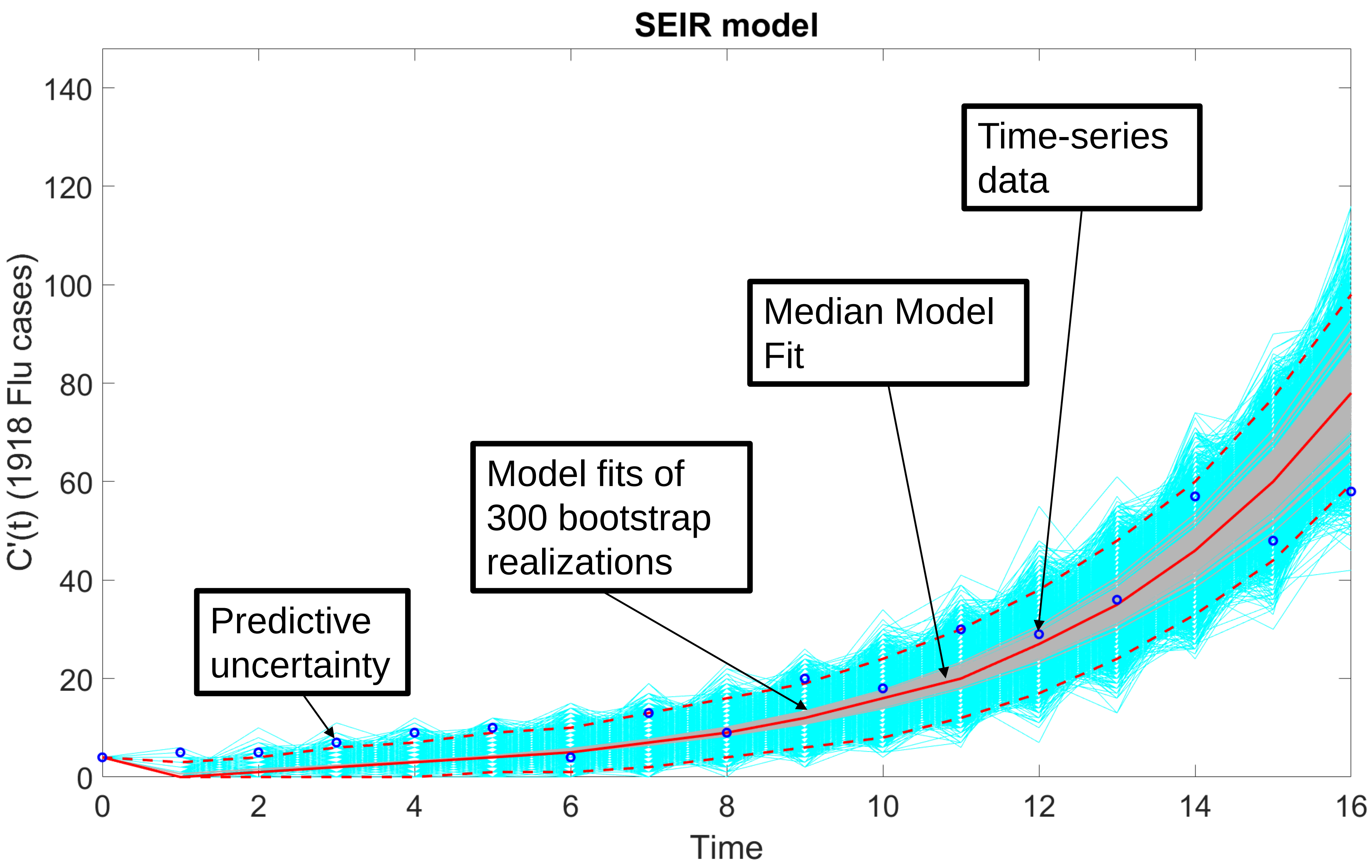
```
plotFit_ODEModel(@OptionsFitFileName,tstart1,tend1,windowsize1)
```

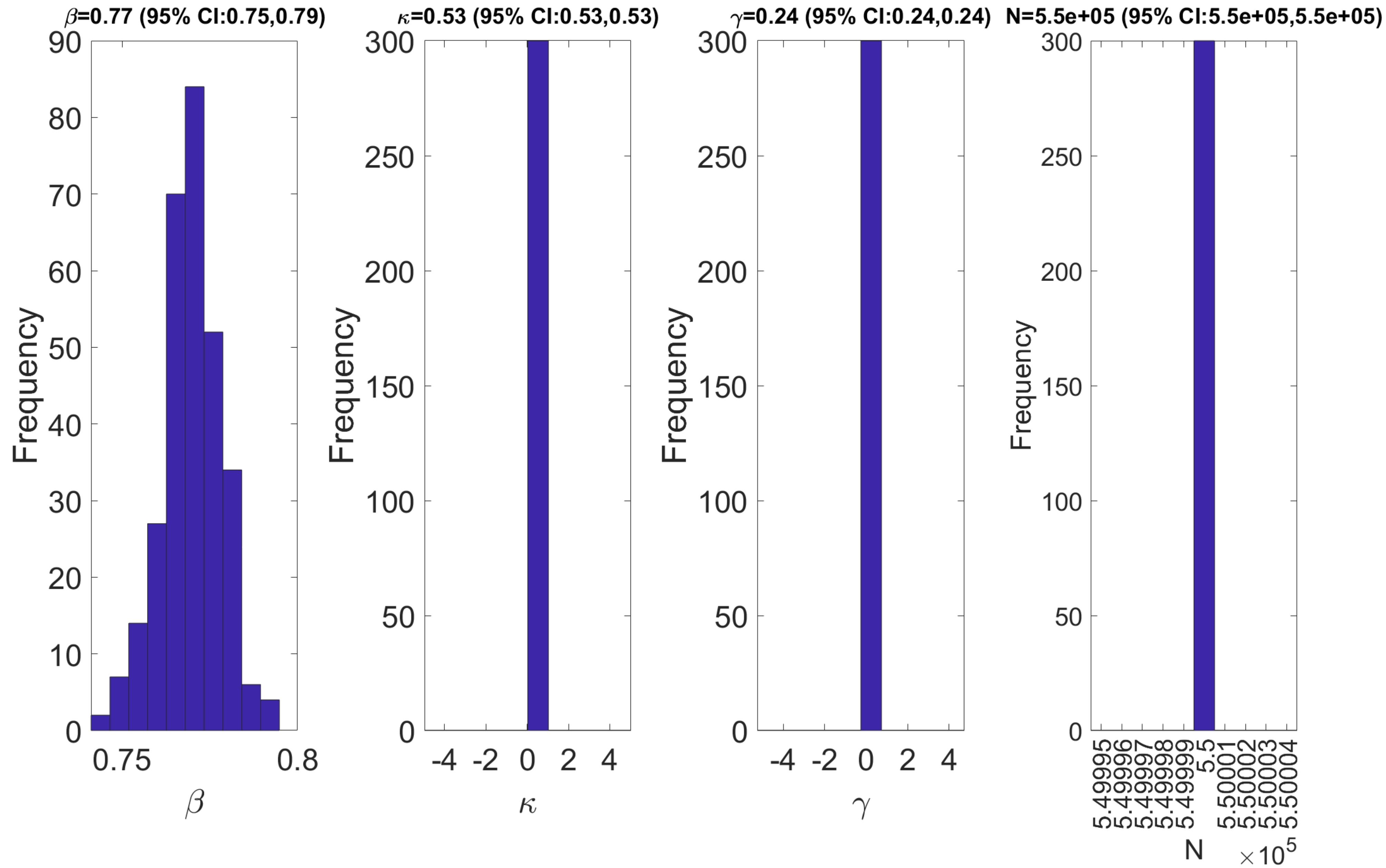
- `tstart1`, `tend1`, and `windowsize1` correspond to the values entered in the rolling window analysis section of the `options_fit.m` file.

Tutorial Code Call: (1) Poisson & (2) Negative Binomial

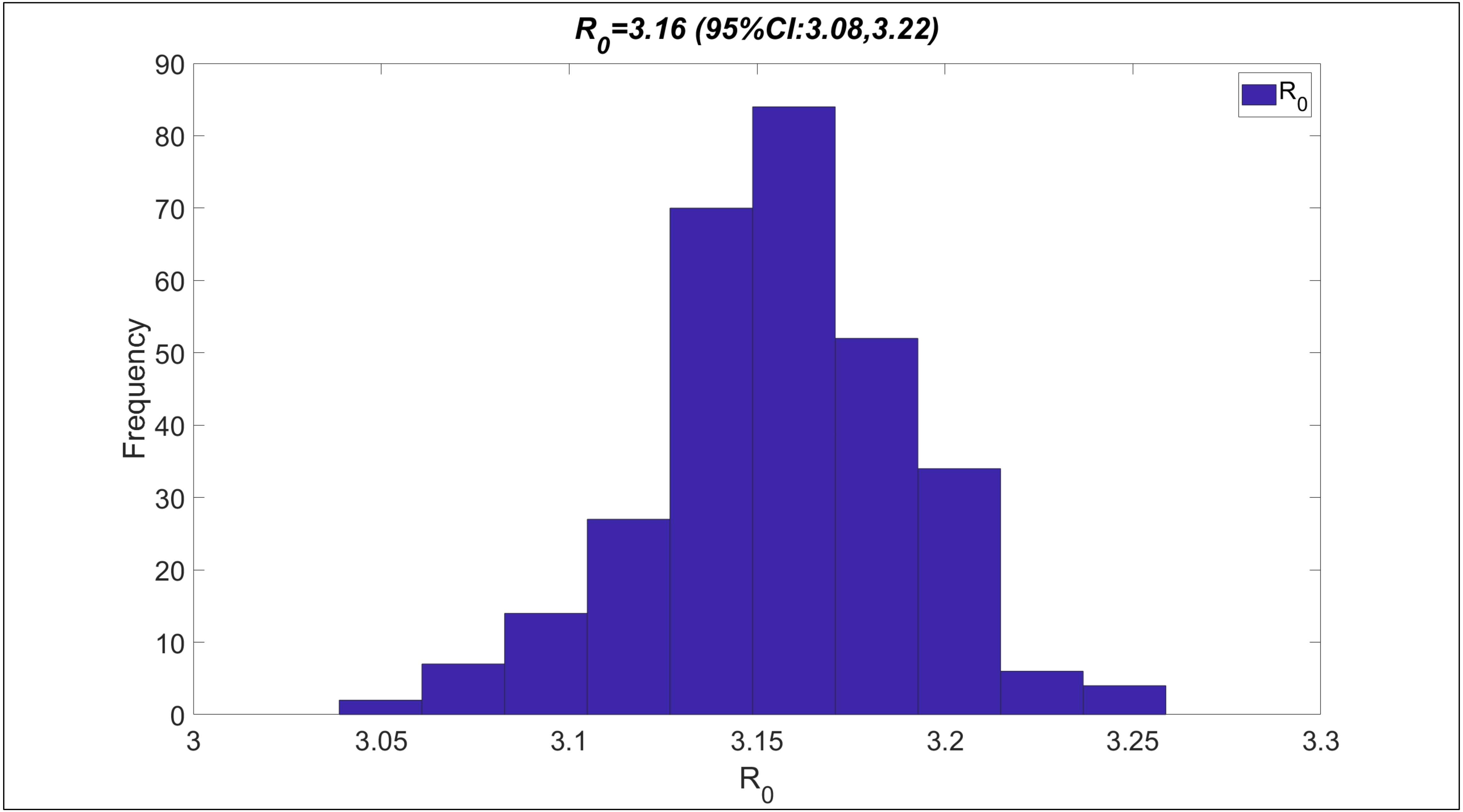
```
(1)plotFit_ODEModel(@options_fit_SEIR_flu1918_dist1_1, 1, 1, 17)
(2)plotFit_ODEModel(@options_fit_SEIR_flu1918_dist1_3, 1, 1, 17)
```

Poisson Output `<dist1 = 1>`

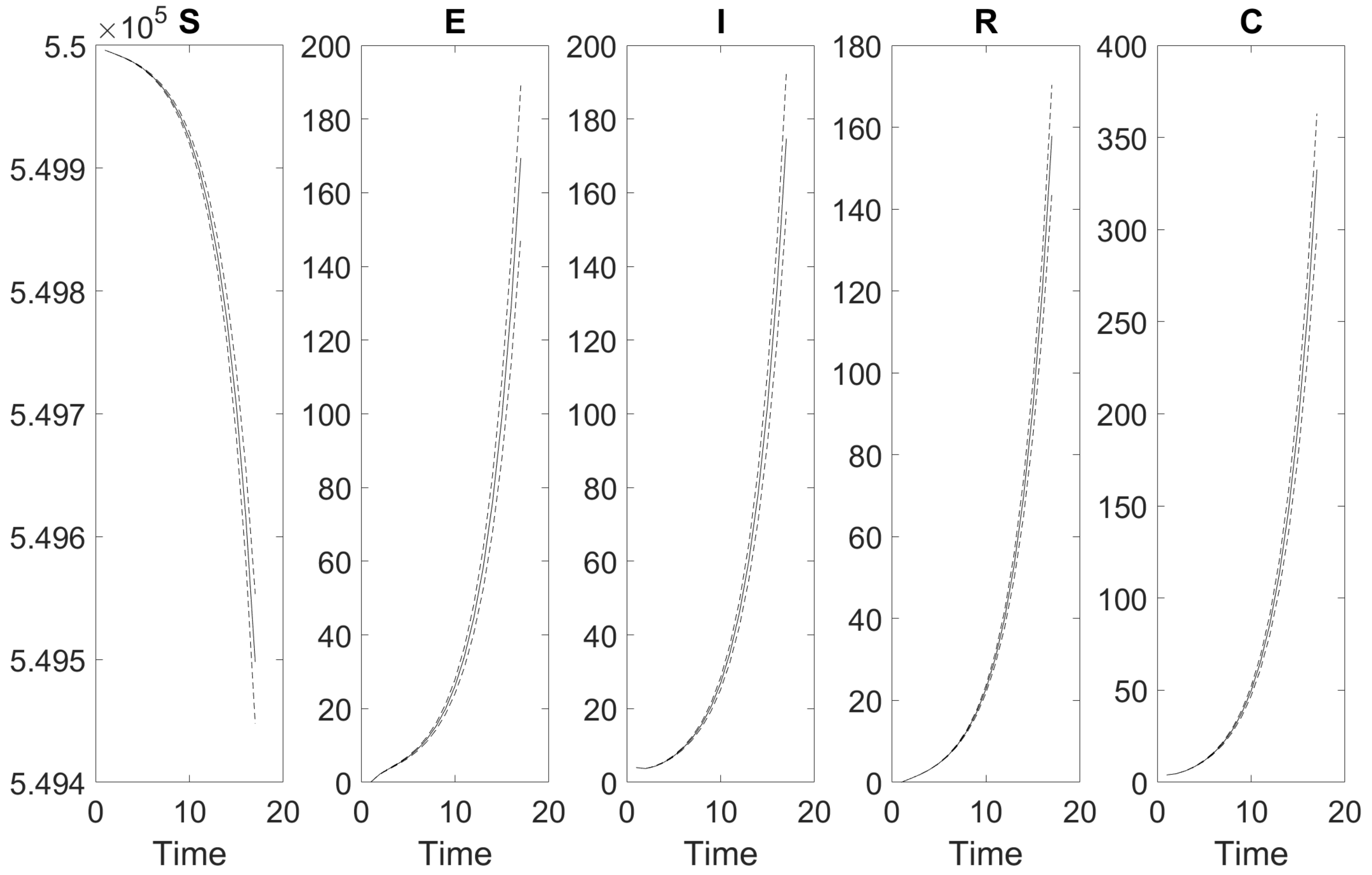

SEIR model
C'(t) (1918 Flu cases)
Time
0
20
40
60
80
100
120
140
0
2
4
6
8
10
12
14
16
Time-series data
Median Model Fit
Model fits of 300 bootstrap realizations
Predictive uncertainty



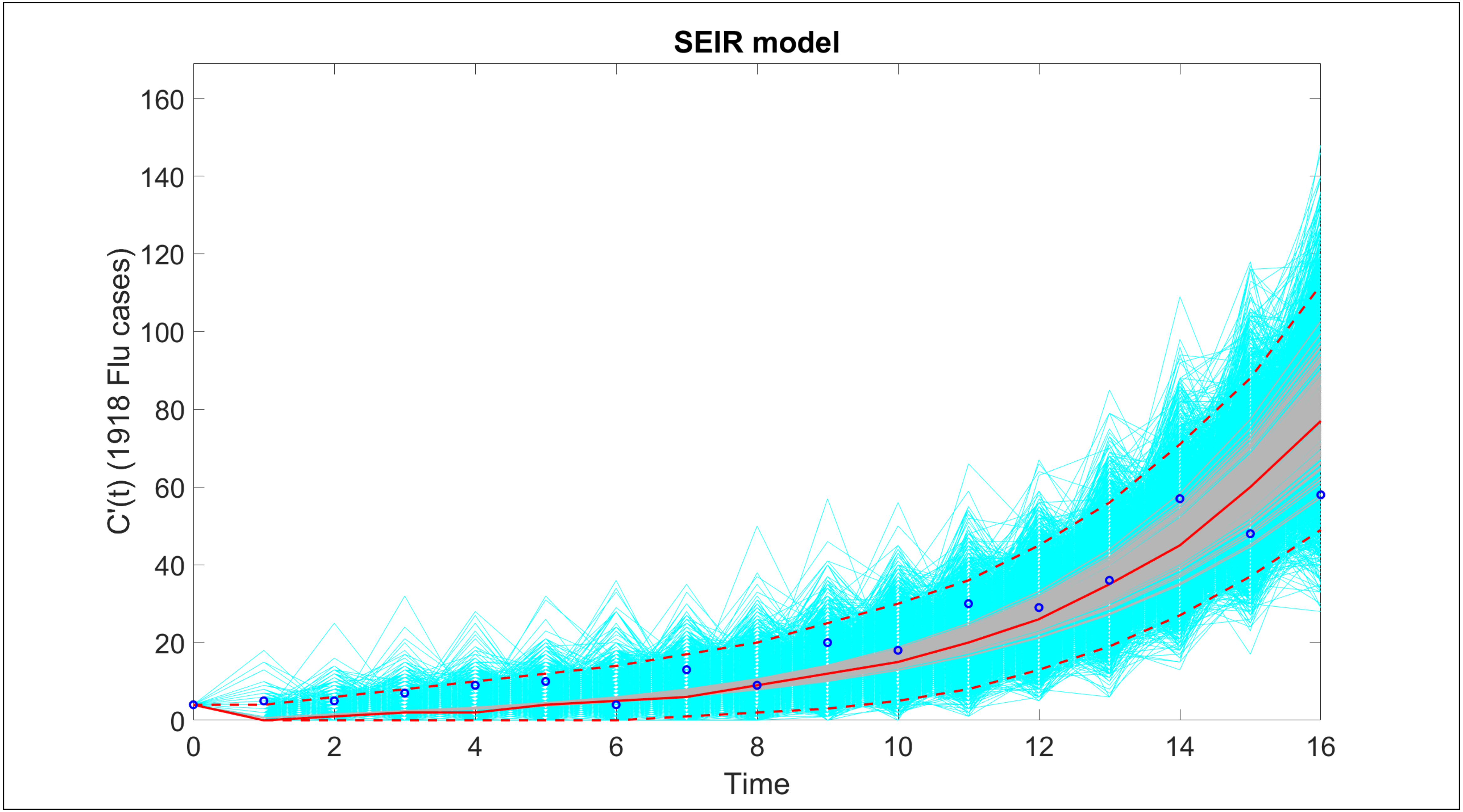
β=0.77 (95% CI:0.75,0.79)
κ=0.53 (95% CI:0.53,0.53)
γ=0.24 (95% CI:0.24,0.24)
N=5.5e+05 (95% CI:5.5e+05,5.5e+05)
Frequency
β
κ
γ
N
$\times 10^5$

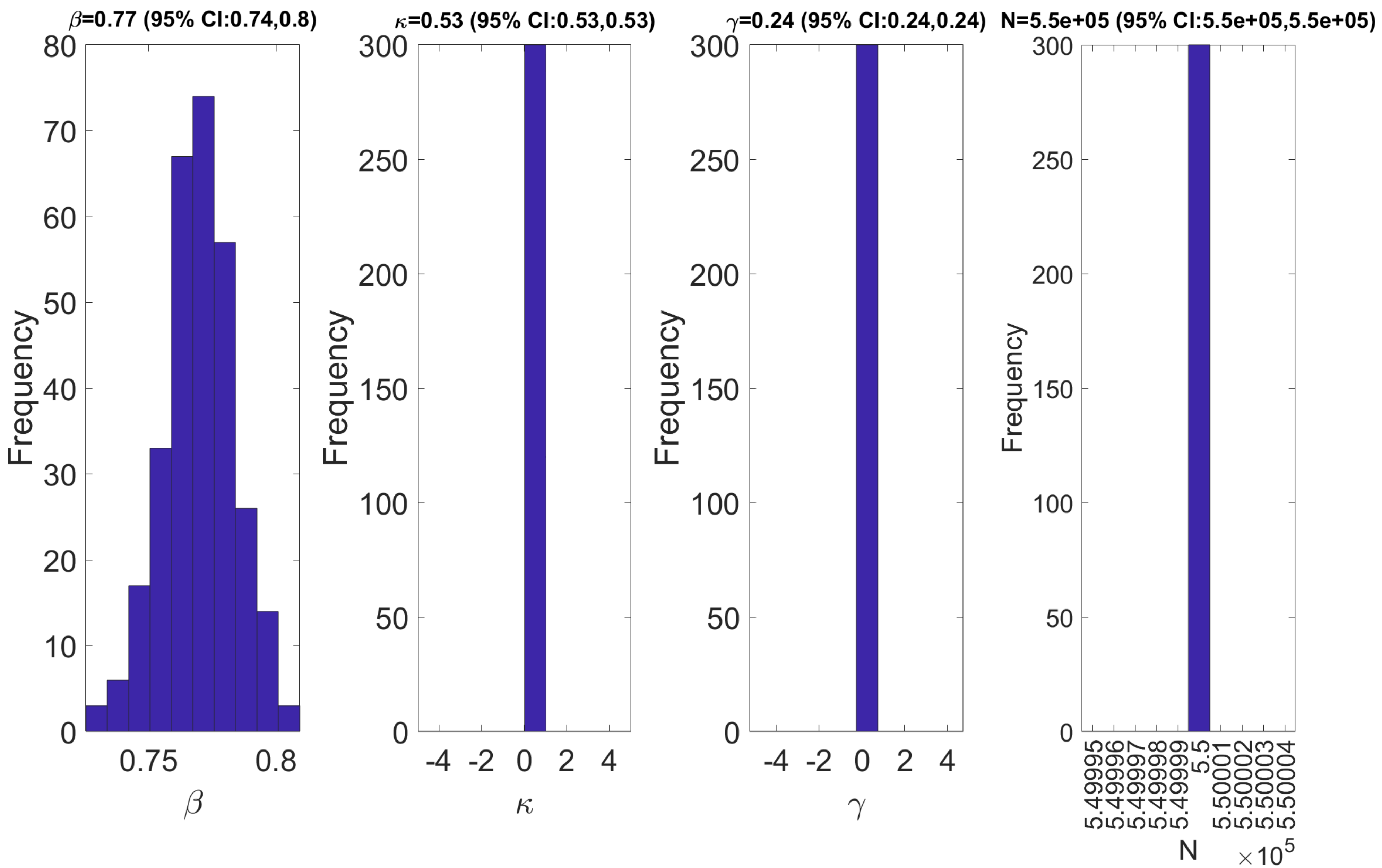


R_0=3.16 (95%CI:3.08,3.22)
Frequency
R_0
R_0
0
10
20
30
40
50
60
70
80
90
3
3.05
3.1
3.15
3.2
3.25
3.3

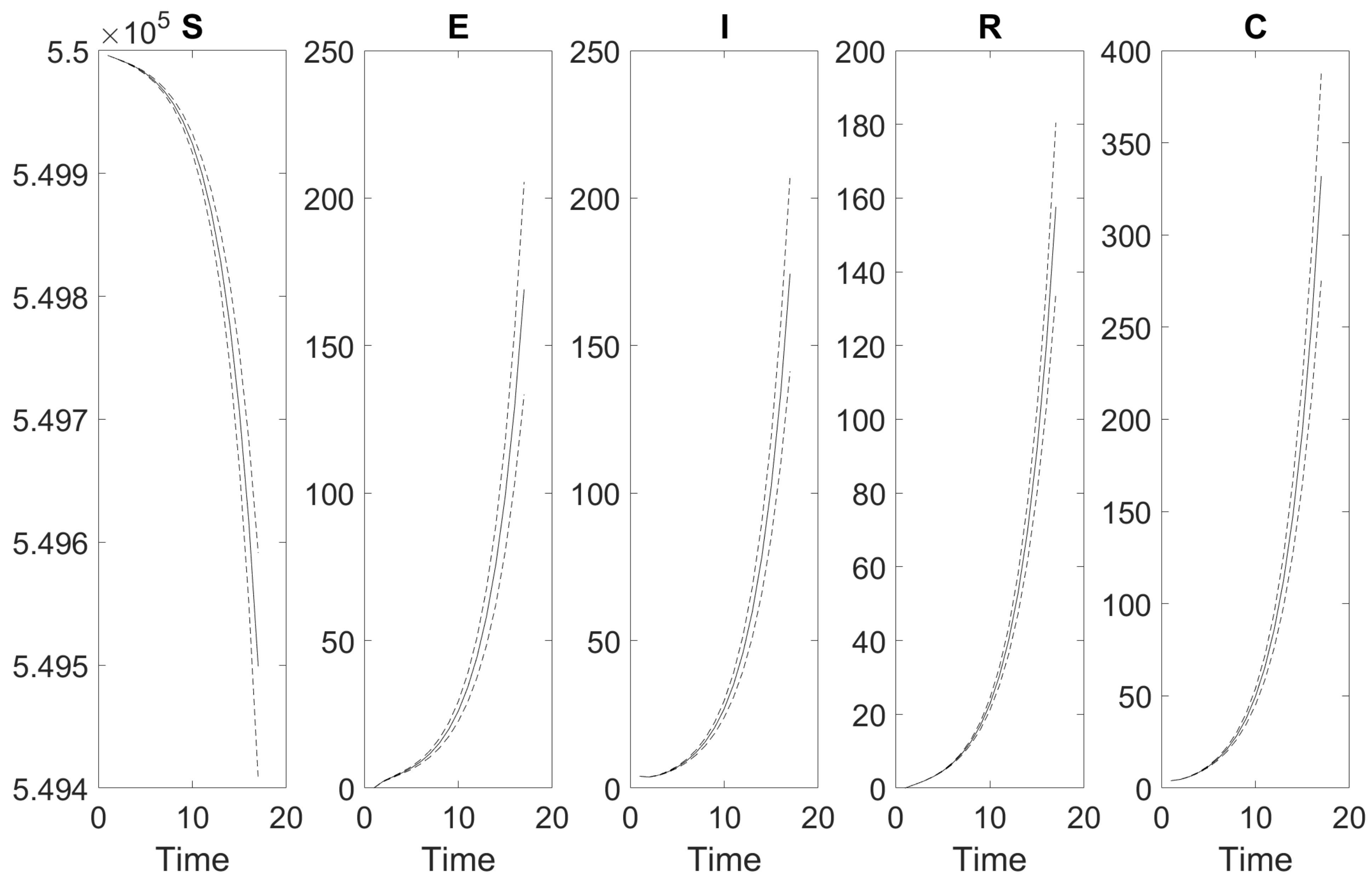

S
E
I
R
C
$\times 10^5$
5.5
5.499
5.498
5.497
5.496
5.495
5.494
200
180
160
140
120
100
80
60
40
20
0
400
350
300
250
150
50
0
10
20
Time

Neg. Binomial Output `<dist1 = 3>`

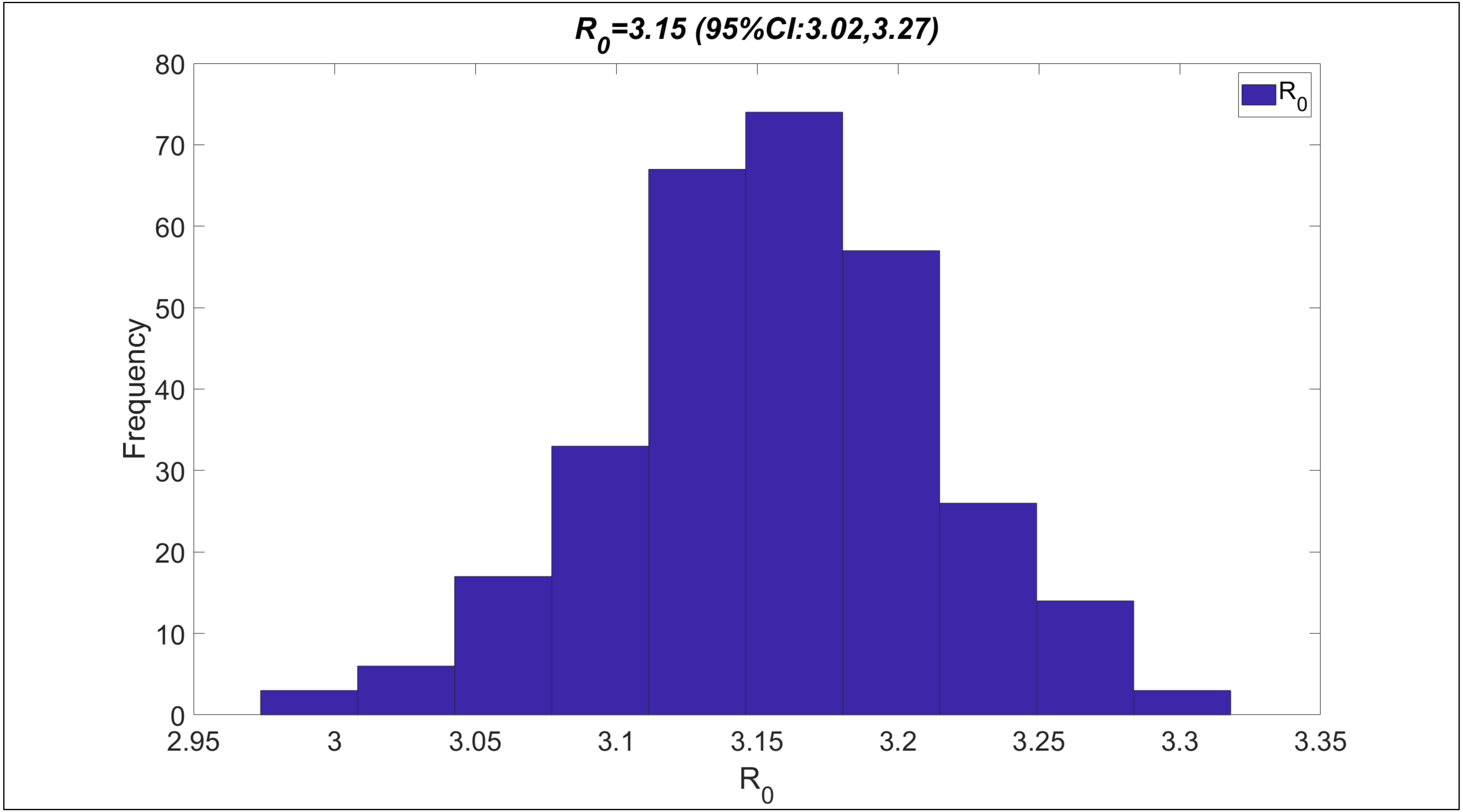

SEIR model
C'(t) (1918 Flu cases)
Time
0
20
40
60
80
100
120
140
160
0
2
4
6
8
10
12
14
16



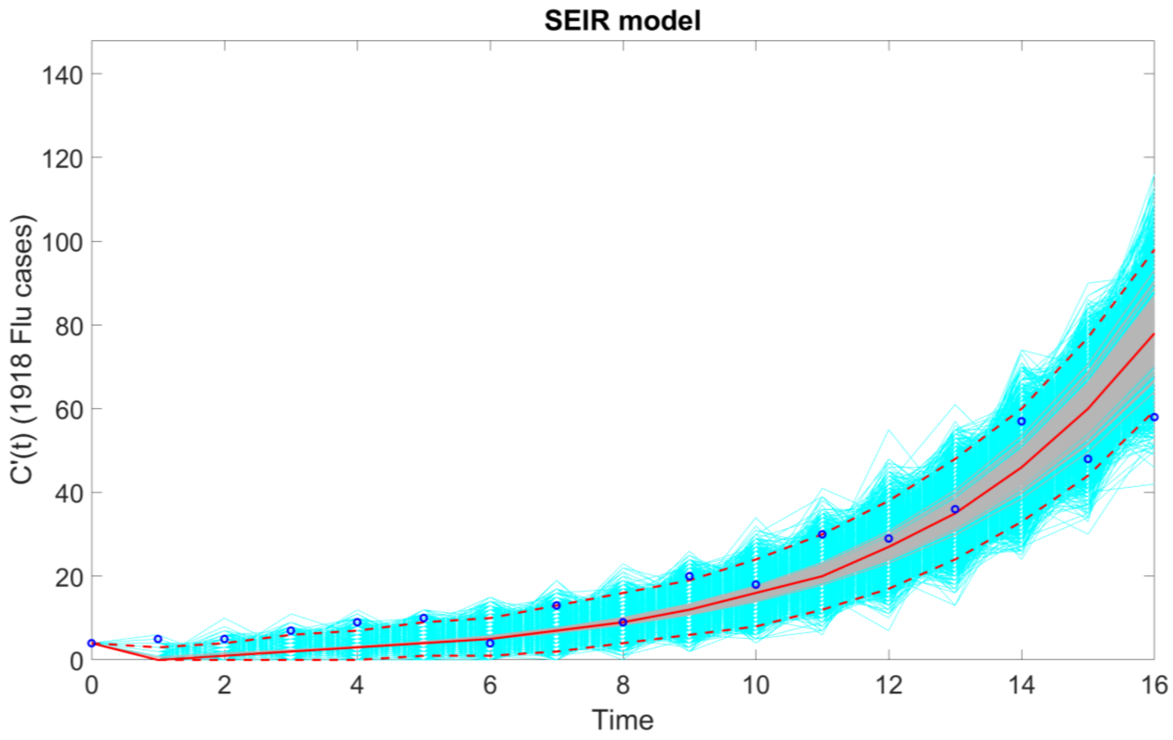
β=0.77 (95% CI:0.74,0.8)
κ=0.53 (95% CI:0.53,0.53)
γ=0.24 (95% CI:0.24,0.24)
N=5.5e+05 (95% CI:5.5e+05,5.5e+05)
Frequency
β
κ
γ
N
×10^5



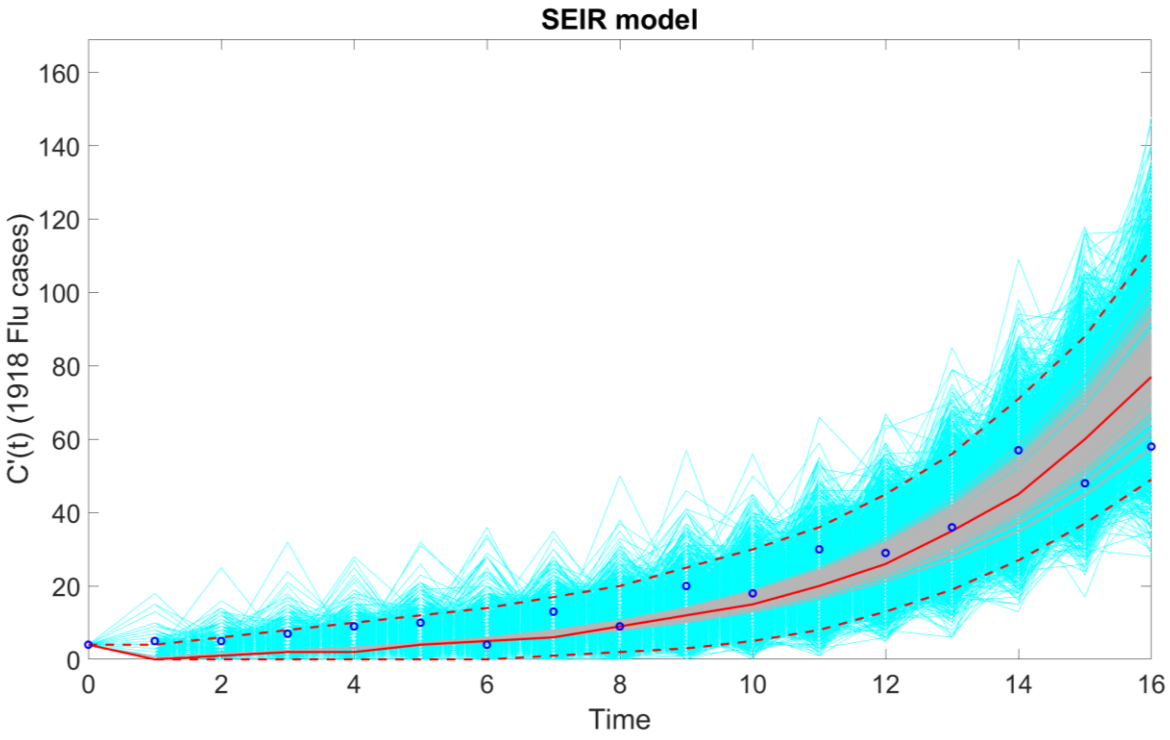
S
E
I
R
C
$\times 10^5$
5.5
5.499
5.498
5.497
5.496
5.495
5.494
250
200
150
100
50
0
250
200
150
100
50
0
200
180
160
140
120
100
80
60
40
20
0
400
350
300
250
200
150
100
50
0
0
10
20
Time


R0=3.15 (95%CI:3.02,3.27)
Frequency
R0
0
10
20
30
40
50
60
70
80
2.95
3
3.05
3.1
3.15
3.2
3.25
3.3
3.35

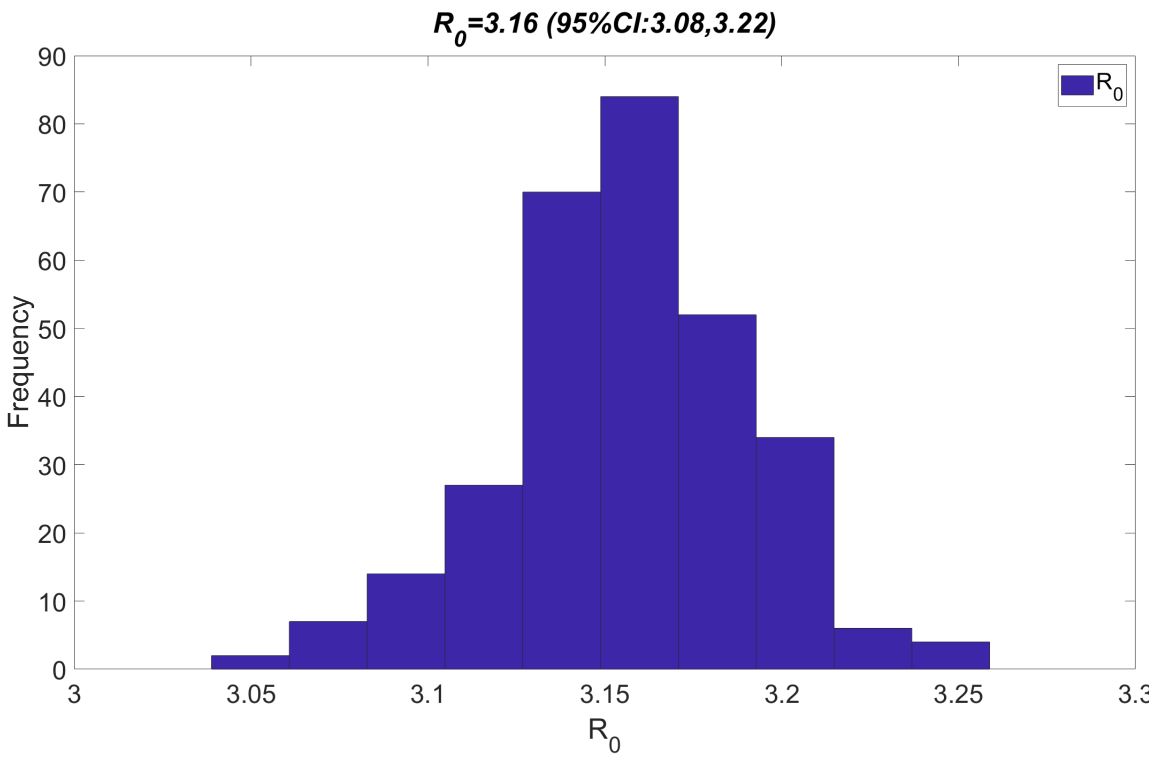
Model Fit Comparison



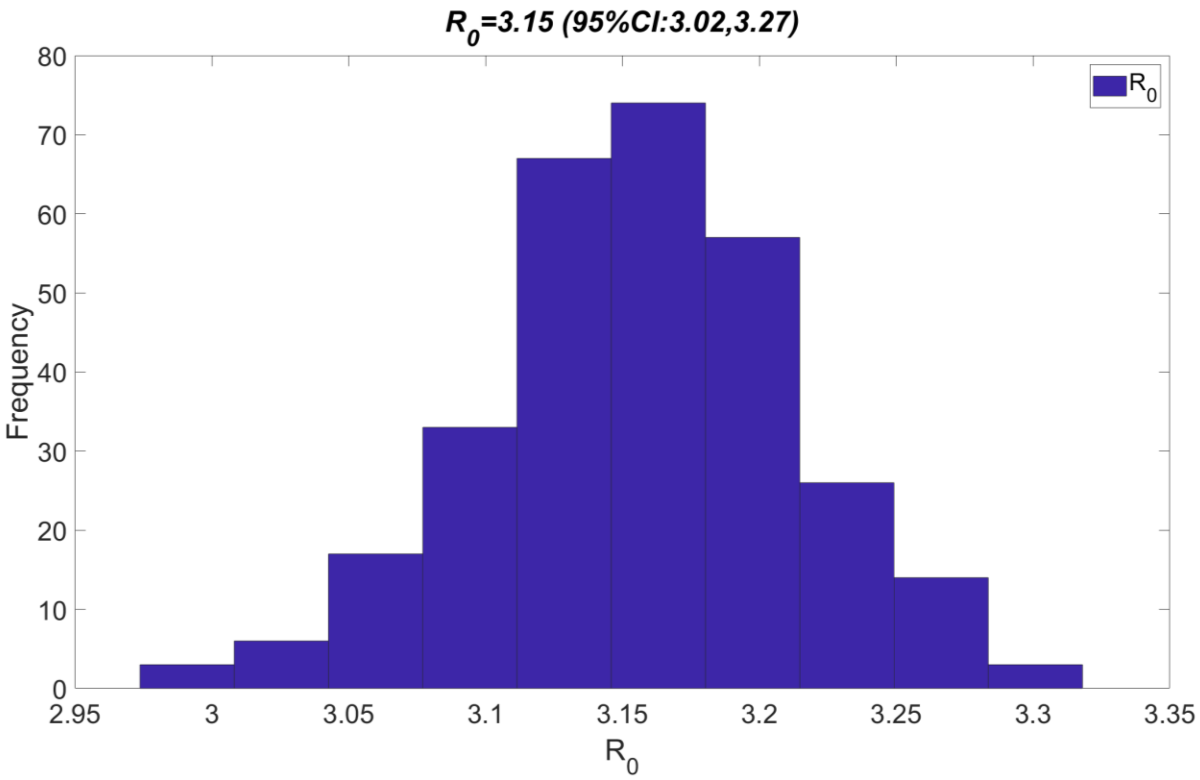
Poisson



Neg. Binomial



Poisson



Neg. Binomial

Model	MAE	MSE	Coverage 95% PI	WIS
SEIR model with Poisson error structure (<dist1>=1)	**5.72**	58.59	58.82	3.87
SEIR model with negative binomial error structure (<dist1>=3)	5.73	**58.21**	**94.12**	**3.67**

* Obtained from *.csv* performance-calibration... files

Generating, plotting and assessing model-based forecasts

Neg. Binomial Error Structure & Exponential Model

Step One: Preparing the Code

```
% <==========================================================================>
% <========================= Forecasting parameters =====================================>
% <==========================================================================>

getperformance=1; % flag or indicator variable (1/0) to calculate forecasting performance or not

forecastingperiod=10; % forecast horizon (number of time units ahead)
```

(1) `getperformance`: Indicator for calculating forecast performance metrics.

- If the forecasting period extends past the available data, getperformance must be marked zero, or an error will occur.

(2) `forecastingperiod`: The forecasting horizon (i.e., how many time points ahead the toolbox should predict)

<u>File Name:</u> `options_forecast_SEIR_flu1918_dist1_3`

Step Two: Generating Forecasts

- Prior to obtaining model fits, forecasts, and associated evaluation criteria we must generate forecasts using the following code call:

```
Run_Forecasting_ODEModels(@OptionsForecastFileName,tstart1,tend1
,windowsize1, forecastingperiod)
```

- `tstart1`, `tend1`, `windowsize1`, and `forecastingperiod` correspond to the values entered in the rolling window analysis section of the `options_forecast.m` file.

Tutorial Code: (1) SEIR Negative Binomial

```
(1)Run_Forecasting_ODEModel(@options_forecast_SEIR_flu1918_dist1_3,
1, 1, 17,10)
```

Step Three: Plotting Forecasts

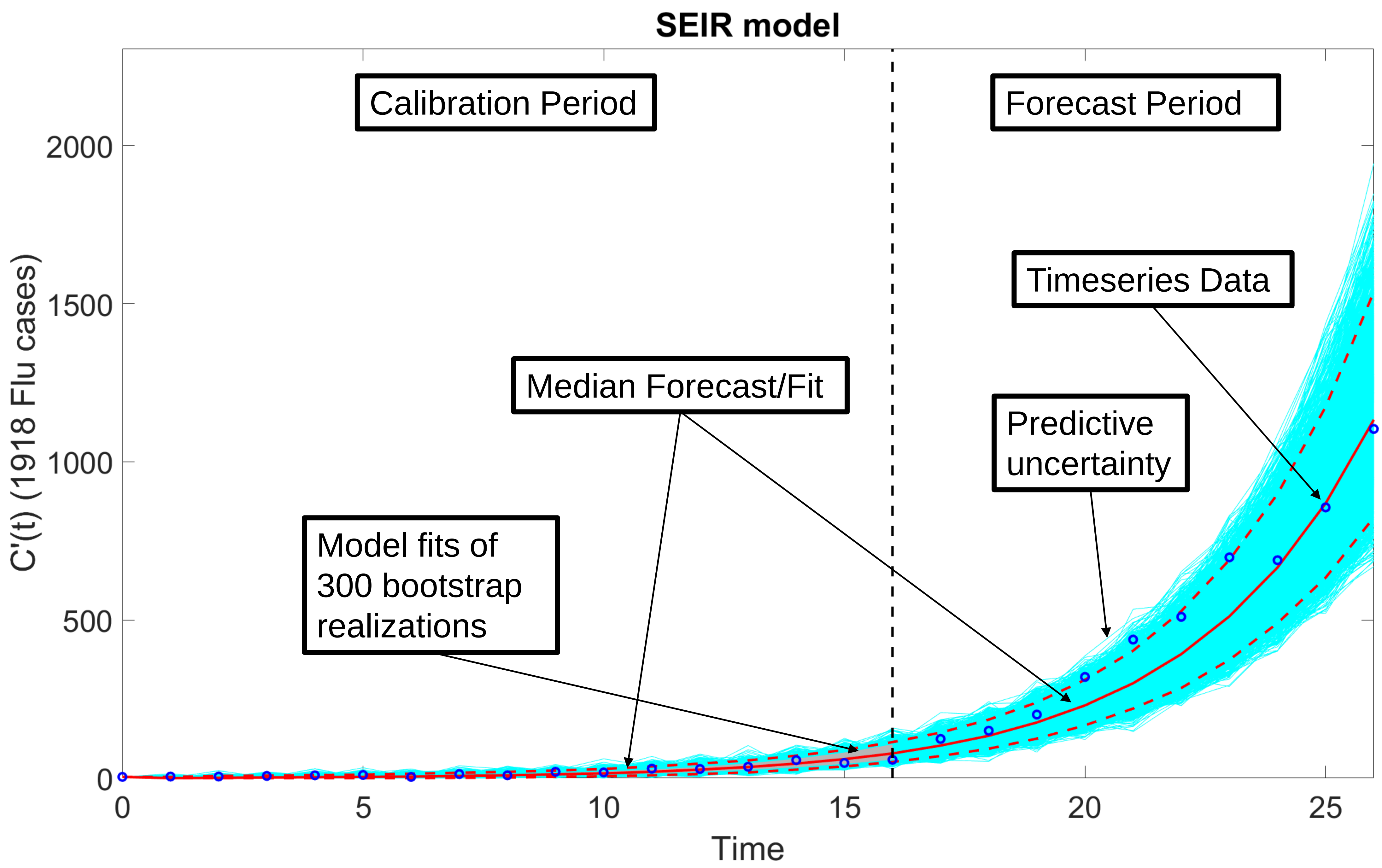
- After generating the model forecasts, we can obtain information related to: (1) forecasts, (2) model parameter estimates, (3) Monte Carlo standard errors, (4) AICc values, (5) calibration performance metrics, (6) forecast performance metrics, & (7) composite parameter information for the specified model and data using the following call (.csv files in output)

```
plotForecast_ODEModel(@OptionsForecastFileName,tstart1,tend1,windowsize1, forecastingperiod)
```

- `tstart1`, `tend1`, `windowsize1`, and `forecastingperiod` correspond to the values entered in the rolling window analysis section of the `options_forecast.m` file.

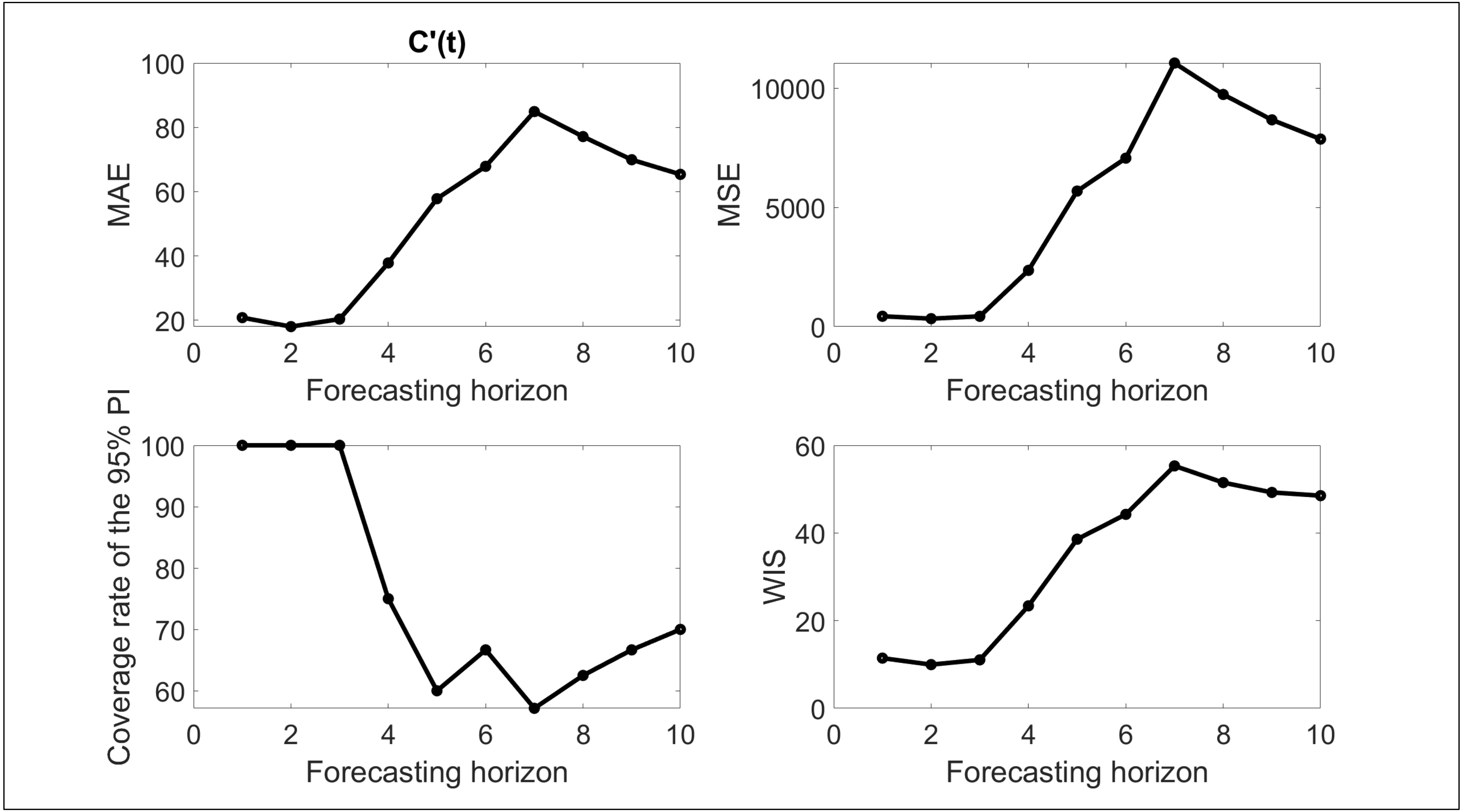
Tutorial Code: (1) SEIR Negative Binomial

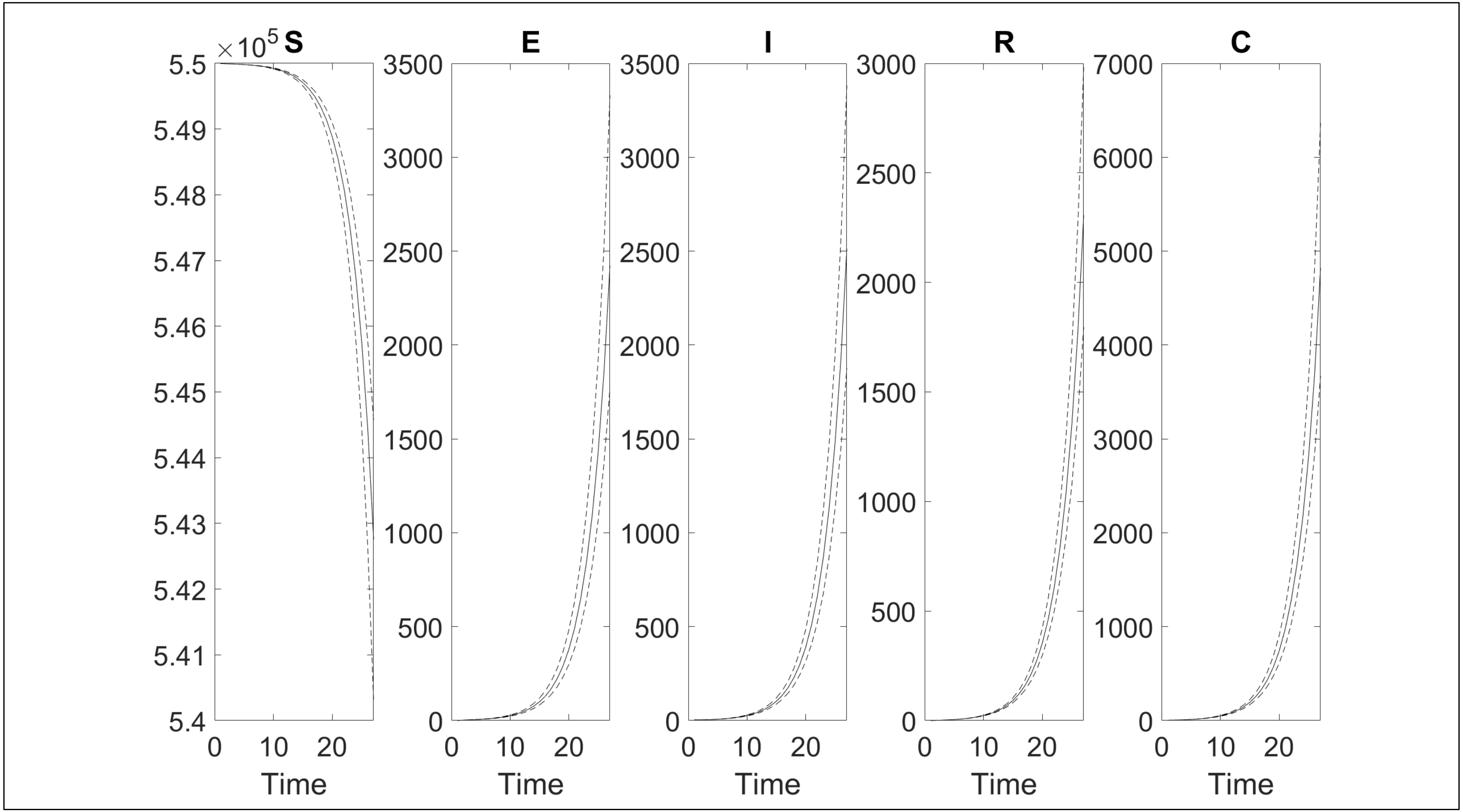
```
(1) plotForecast_ODEModel(@options_forecast_SEIR_flu1918_dist1_3, 1, 1, 17,10)
```


SEIR model
Calibration Period
Forecast Period
Timeseries Data
Median Forecast/Fit
Predictive uncertainty
Model fits of 300 bootstrap realizations
C'(t) (1918 Flu cases)
Time
0
500
1000
1500
2000
0
5
10
15
20
25

Forecast-model_name-SEIR model-vars.fit_index-5-tstart-1-fixI0-1-method-3-dist-3-tstart-1-tend-1-calibrationperiod-17-horizon-10-1918 Flu-c... • Saved to this PC

time	data	median	LB	UB
0	4	4	4	4
1	5	0	0	4
2	5	1	0	7
3	7	2	0	8
4	9	3	0	10
5	10	4	0	11.5
6	4	5	0	14
7	13	7	1	16.5
8	9	9	2	20
9	20	11	3	24
10	18	16	5.5	29
11	30	20	8	36
12	29	26	13	45
13	36	35	19	56
14	57	46	27	70
15	48	60	37	88
16	58	78	50	111
17	124	102	68	143
18	150	133	93	185
19	201	174	122	237
20	320	227	162	307
21	438	299	213	401.5
22	510	390	281	520
23	698	508	362	686
24	689	663	475.5	899
25	856	864	620	1171
26	1104	1126	804	1546


C'(t)
MAE
MSE
Coverage rate of the 95% PI
WIS
Forecasting horizon
Forecasting horizon
Forecasting horizon
Forecasting horizon



$\times 10^5$
S
E
I
R
C
5.5
5.49
5.48
5.47
5.46
5.45
5.44
5.43
5.42
5.41
5.4
3500
3000
2500
2000
1500
1000
500
0
7000
6000
5000
4000
0
10
20
Time

Model Comparison

SEIR Neg. Binomial & Exponential Neg. Binomial

Specifying the Exponential Model

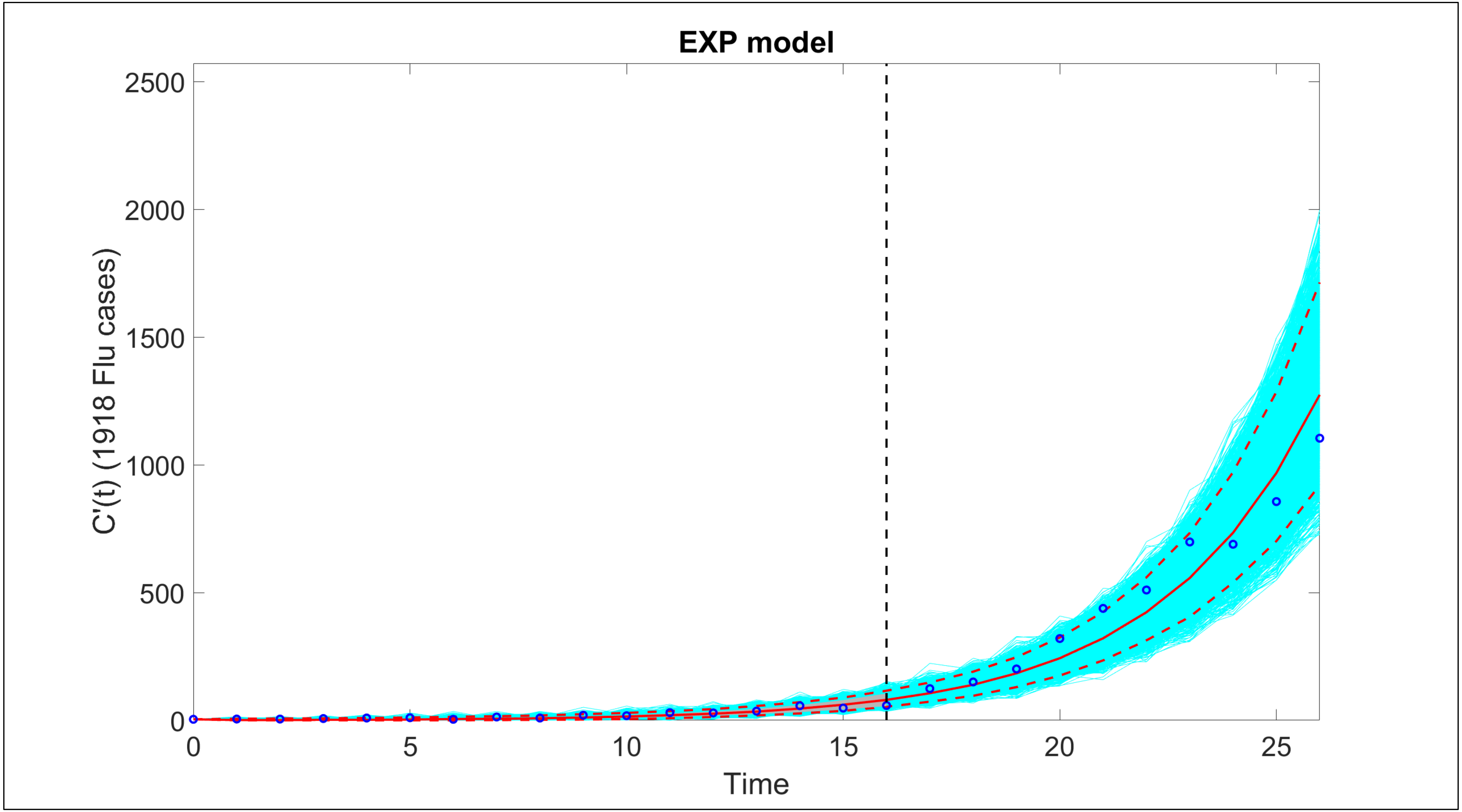
```
% <==========================================
% < Author: Gerardo Chowell  ===================
% <==========================================

function dx=EXP(t,x,params0)

% parameters in order: r

dx=zeros(1,1);

dx(1,1)=params0(1)*x(1,1);
```

$$C'(t) = r * C(t)$$

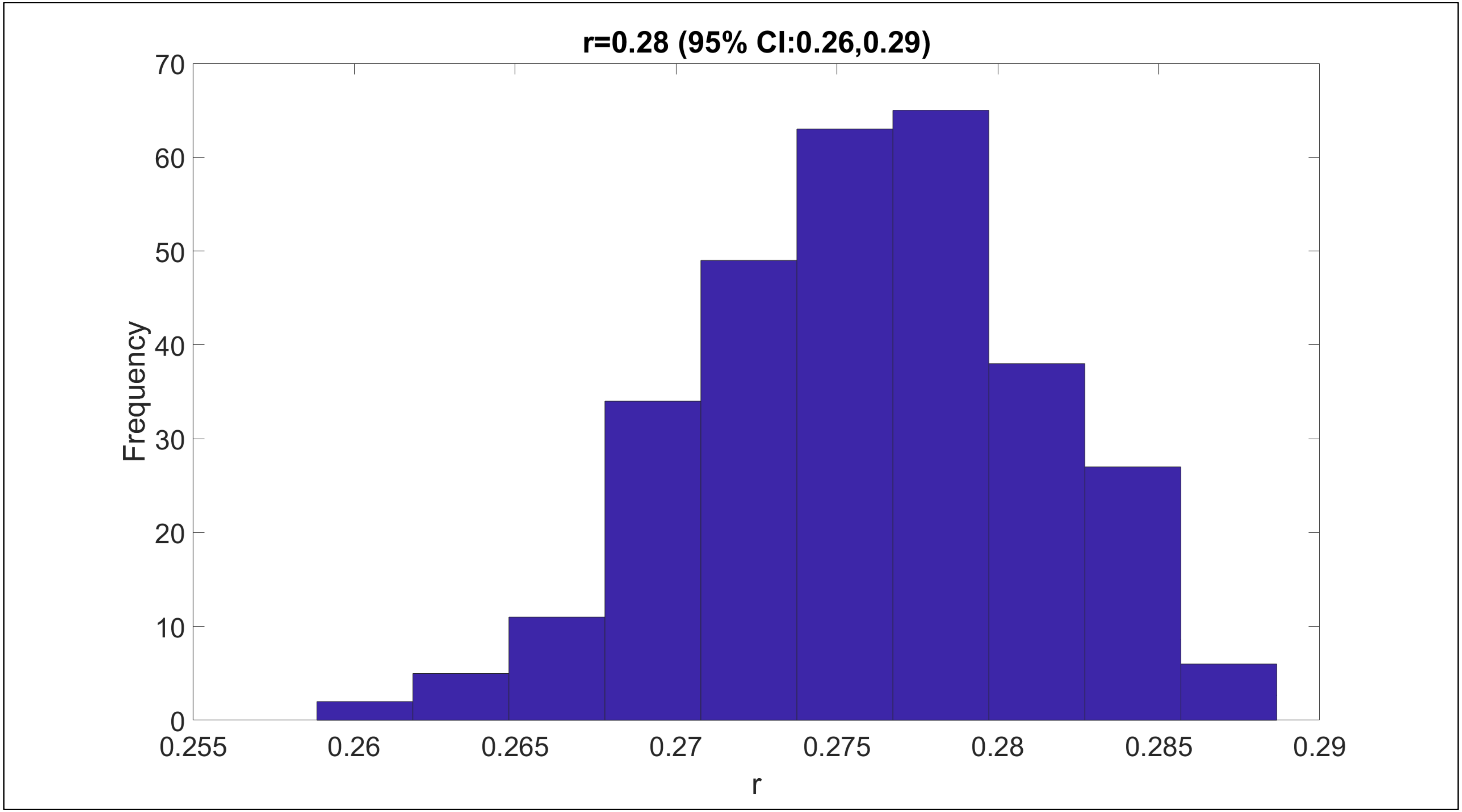
r : Growth rate $(r > 0)$

```
Run_Forecasting_ODEModel(@options_forecast_EXP_flu1918_dist1_3, 1, 1, 17)
```

```
% <=============================================================================>
% <============================== ODE model ======================================>
% <=============================================================================>

model.fc=@EXP;  % name of the model function
model.name='EXP model'; % string indicating the name of the ODE model

params.label={'r'}; % list of symbols to refer to the model parameters
params.LB=[0];  % lower bound values of the parameter estimates
params.UB=[10]; % upper bound values of the parameter estimates
params.initial=[0.18]; % initial parameter values/guesses
params.fixed=[0]; % Boolean vector to indicate any parameters that should remain fixed (1) to initial values indicated in par
params.fixI0=1; % Boolean variable indicating if the initial value of the fitting variable is fixed according to the first ob
params.composite='';  % Estimate a composite function of the individual model parameter estimates otherwise it is left empty.
params.composite_name=''; % Name of the composite parameter
params.extra0=[]; % used to pass pass any extra parameters (e.g., data, static variables) to the model function

vars.label={'C'}; % list of symbols to refer to the variables included in the model
vars.initial=5; % vector of initial conditions for the model variables
vars.fit_index=1; % index of the model's variable that will be fit to the observed time series data
vars.fit_diff=1; % boolean variable to indicate if the derivative of model's fitting variable should be fit to data.
```



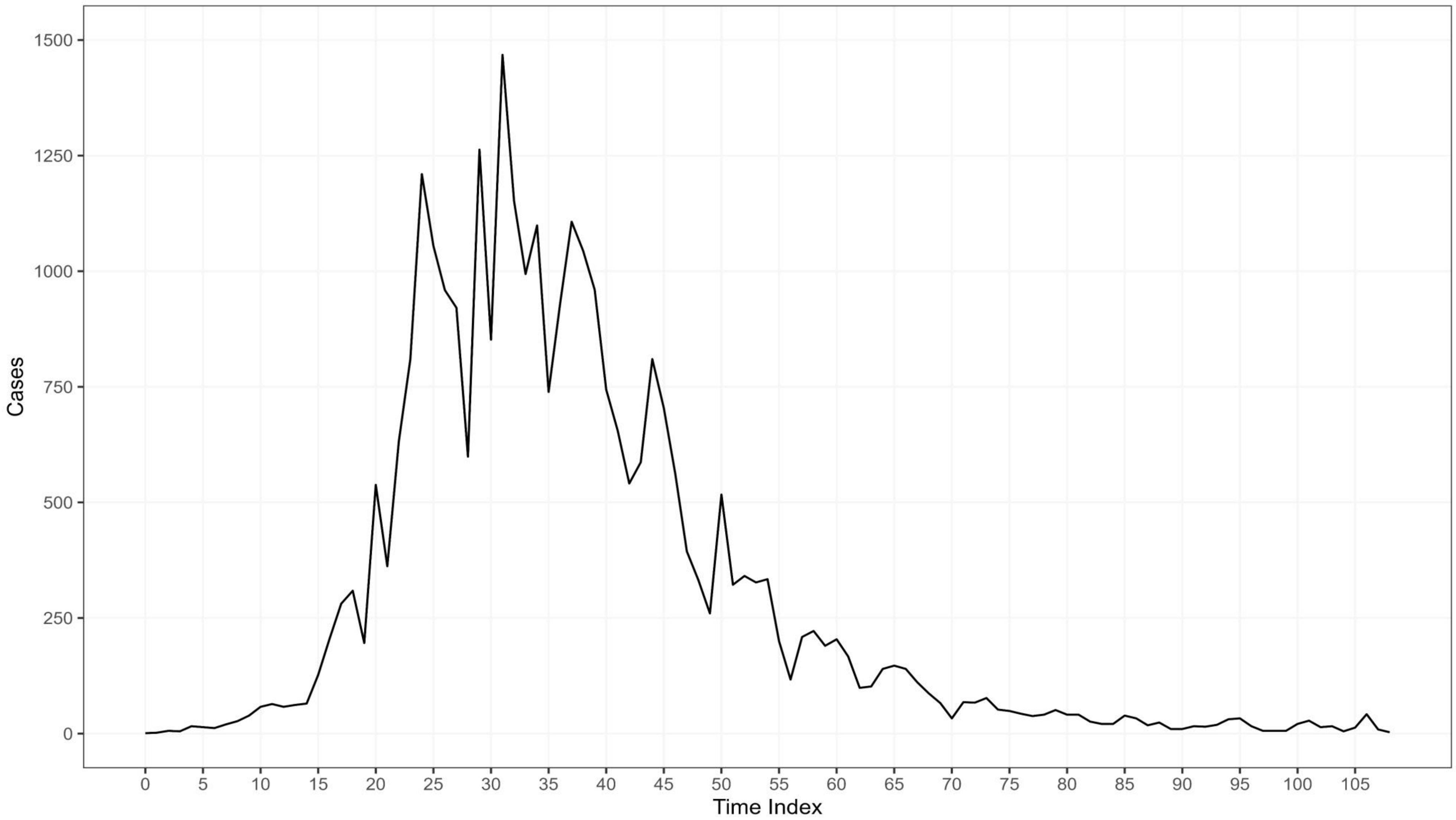
EXP model
C'(t) (1918 Flu cases)
2500
2000
1500
1000
500
0
0
5
10
15
20
25
Time



r=0.28 (95% CI:0.26,0.29)
Frequency
0
10
20
30
40
50
60
70
0.255
0.26
0.265
0.27
0.275
0.28
0.285
0.29
r

Model	MAE	MSE	Coverage 95% PI	WIS
Calibration Performance				
SEIR model with NB error structure (<dist1>=3)	**5.74**	**59.51**	**94.12**	**3.66**
Exponential model with negative binomial error structure (<dist1>=3)	5.99	67.03	100.0	3.84
Forecast Performance				
SEIR model with NB error structure (<dist1>=3)	**65.48**	**8058.65**	70.0	49.80
Exponential model with negative binomial error structure (<dist1>=3)	79.79	9252.18	**90.0**	**46.72**

Tutorial #2: Swiss COVID-19



Daily number of reported cases of SARS-CoV-2 infection in Switzerland reported at the national level during the first wave (Feb. 2020 to June 2020).

Specifying `SEIURC` and `R0`

Early Phase

```matlab
% <==========================================================================>
% < Author: Gerardo Chowell  ================================================>
% <==========================================================================>

function dx=SEIRUnder(t,x,params0,extra0)

beta0=params0(1);
alpha=params0(2); % non-homogenous mixing parameter
rho=params0(3);
k=params0(4);
gamma1=params0(5);
N=params0(6);


dx=zeros(6,1);  % define the vector of the state derivatives: S, E, I, U, R, C

dx(1,1)= -beta0*x(1,1).*((x(3,1)+x(4,1)).^alpha)./N; %S

dx(2,1)= beta0*x(1,1).*((x(3,1)+x(4,1)).^alpha)./N - k*x(2,1); %E

dx(3,1)= k*rho*x(2,1) - gamma1*x(3,1); %I

dx(4,1)= k*(1-rho)*x(2,1) - gamma1*x(4,1); %U

dx(5,1)= gamma1*(x(3,1)+x(4,1)); %R

dx(6,1)= k*rho*x(2,1); %C
```

$$\begin{cases} \dot{S} = -\beta S(t)\dfrac{I(t)+U(t)^{\alpha}}{N} \\ \dot{E} = \beta S(t)\dfrac{I(t)+U(t)^{\alpha}}{N} - \kappa E(t) \\ \dot{I} = \kappa\rho E(t) - \gamma I(t) \\ \dot{U} = \kappa(1-\rho)E(t) - \gamma U(t) \\ \dot{R} = \gamma(I(t)+U(t)) \\ \dot{C} = \kappa\rho E(t) \end{cases}$$

```
% <==============================================================================>
% <============================== ODE model ==========================================>
% <==============================================================================>

model.fc=@SEIR_unreported; % name of the model function
model.name='SEIR swiss_covid underreporting';   % string indicating the name of the ODE model

params.label={'\beta_0','\alpha','\rho','\kappa','\gamma','N'};  % list of symbols to refer to the model parameters
params.LB=[0.001 0.5 0.01 0.01 0.01 47332614]; % lower bound values of the parameter estimates
params.UB=[5 1 1 1 1 47332614]; % upper bound values of the parameter estimates
params.initial=[0.6 1 1 1/5 1/4 47332614]; % initial parameter values/guesses
params.fixed=[0 1 0 1 1 1]; % Boolean vector to indicate any parameters that should remain fixed (1) to initial values ir
params.fixI0=1; % Boolean variable indicating if the initial value of the fitting variable is fixed according to the firs
params.composite=@R0s_unreported;  % Estimate a composite function of the individual model parameter estimates otherwise
params.composite_name='R0s_unreported'; % Name of the composite parameter
params.extra0='';

vars.label={'S','E','I','U','R','C'}; % list of symbols to refer to the variables included in the model
vars.initial=[params.initial(6)-1 0 1 0 0 1];  % vector of initial conditions for the model variables
vars.fit_index=6; % index of the model's variable that will be fit to the observed time series data
vars.fit_diff=1; % boolean variable to indicate if the derivative of model's fitting variable should be fit to data.
```

Preparing `options_fit.m` & `options_forecast.m`

Fitting and forecasting the early growth phase (First 20-days)

```
% <============================================================================>
% <=================== Declare global variables =======================================>
% <============================================================================>

global method1 % Parameter estimation method



% <============================================================================>
% <================================ Datasets properties =======================>
% <============================================================================>
% Located in the input folder, the time series data file is a text file with extension *.txt.
% The time series data file contains the incidence curve of the epidemic of interest.
% The first column corresponds to time index: 0,1,2, ... and the second
% column corresponds to the observed time series data.

cadfilename1='curve-covid-firtstwave-swiss-reporteddt';

caddisease='COVID-19'; % string indicating the name of the disease related to the time series data

datatype='cases'; % string indicating the nature of the data (cases, deaths, hospitalizations, etc)
```

```matlab
% <===================================================================================>
% <============================ Parameter estimation ============================>
% <===================================================================================>

method1=0; % Type of estimation method

% Nonlinear least squares (LSQ)=0,
% MLE Poisson=1,
% MLE (Neg Binomial)=3, with VAR=mean+alpha*mean;
% MLE (Neg Binomial)=4, with VAR=mean+alpha*mean^2;
% MLE (Neg Binomial)=5, with VAR=mean+alpha*mean^d;

dist1=0; % Define dist1 which is the type of error structure. See below:

%dist1=0; % Normal distribution to model error structure (method1=0)
%dist1=1; % Poisson error structure (method1=0 OR method1=1)
%dist1=2; % Neg. binomial error structure where var = factor1*mean where
                % factor1 is empirically estimated from the time series
                % data (method1=0)
%dist1=3; % MLE (Neg Binomial) with VAR=mean+alpha*mean  (method1=3)
%dist1=4; % MLE (Neg Binomial) with VAR=mean+alpha*mean^2 (method1=4)
%dist1=5; % MLE (Neg Binomial)with VAR=mean+alpha*mean^d (method1=5)


switch method1
    case 1
        dist1=1;
    case 3
        dist1=3;
    case 4
        dist1=4;
    case 5
        dist1=5;
end


numstartpoints=2*4^2; % Number of initial guesses for optimization procedure using MultiStart

B=300; % number of bootstrap realizations to characterize parameter uncertainty
```

Tutorial

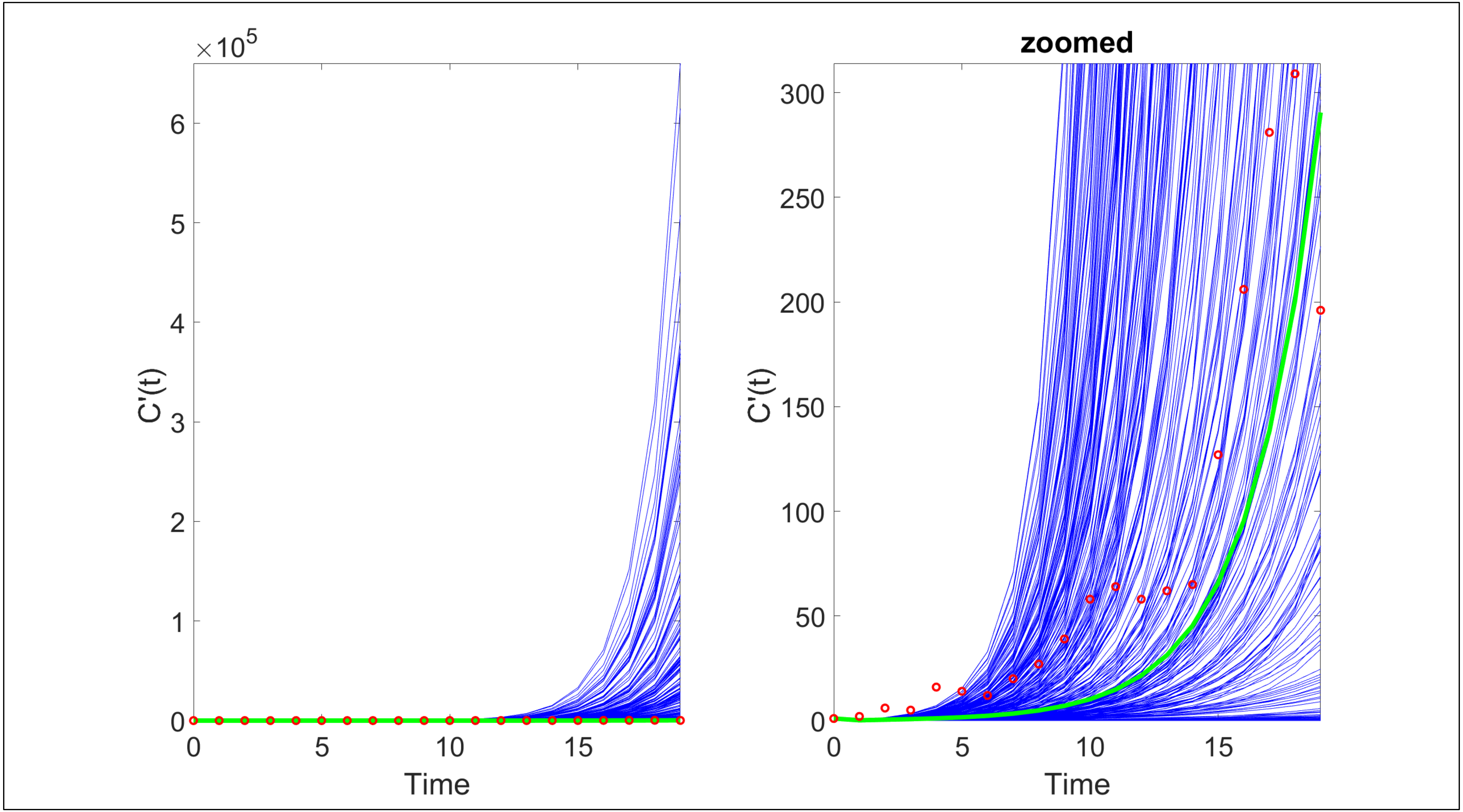
- As the case counts are much higher compared to Tutorial #1, we will be using <u>nonlinear least squares (NLSQ)</u> and assuming a <u>normal error distribution</u>.
- Using 32 start points (`<numstartpoints> = 32`)
- Using 300 bootstrap realizations to characterize parameter uncertainty (`<B> = 300`)

Fitting to the first 20 days

```
% <==============================================================================>
% <========================== Forecasting parameters ===================================>
% <==============================================================================>

getperformance=1; % flag or indicator variable (1/0) to calculate forecasting performance or not

forecastingperiod=10; % forecast horizon (number of time units ahead)

% <==================================================================================>
% <========================== Parameters of the rolling window analysis =========================>
% <==================================================================================>

windowsize1=20;  % moving window size

tstart1=1; % time point for the start of rolling window analysis

tend1=1;  %time point for the end of the rolling window analysis

printscreen1=1;
```

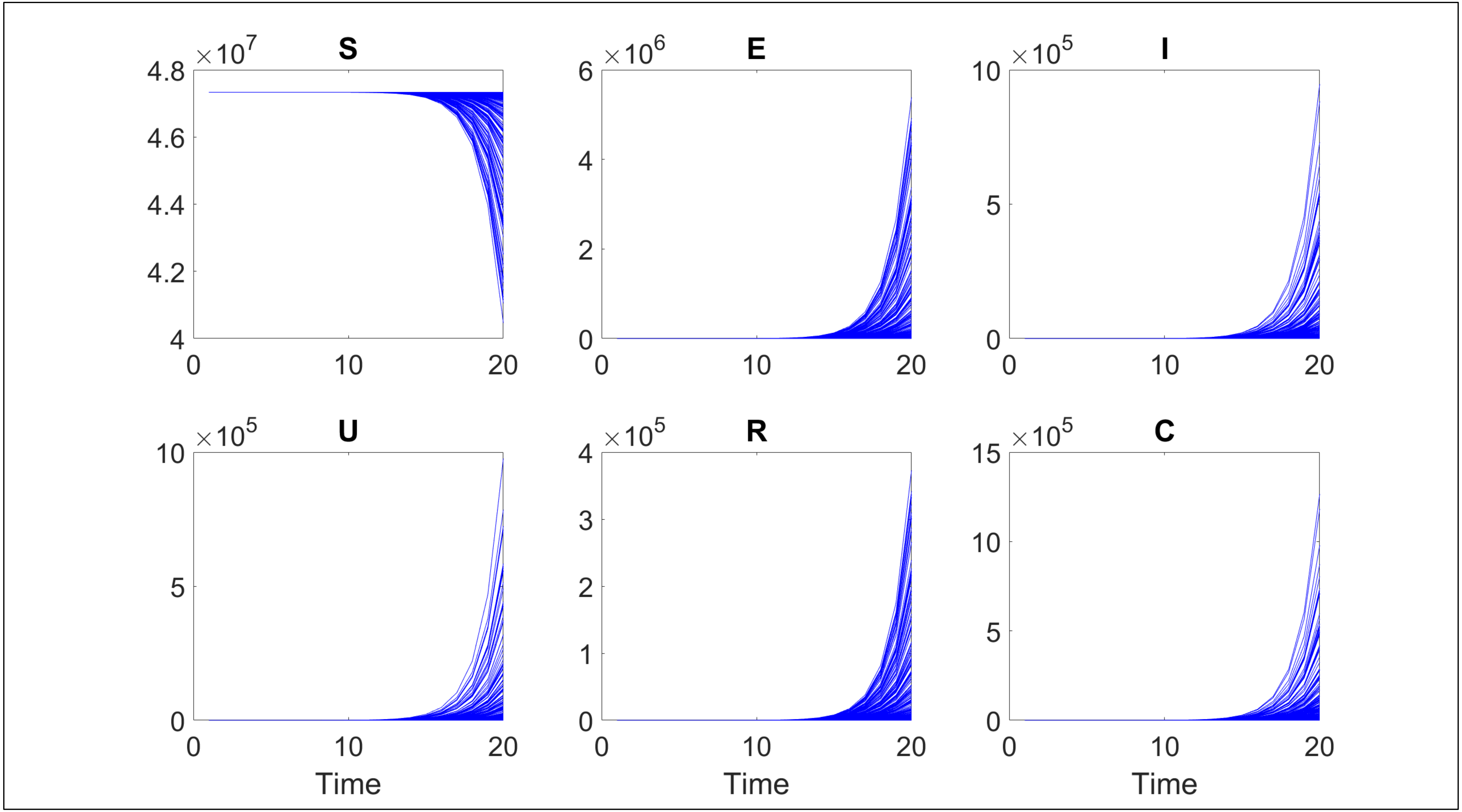
Generating preliminary model solutions

```
plotODEModel(@options_fit_SEIR_unreportedNIn_covid_swiss_dist1_0)
```



×10^5
6
5
4
3
2
1
0
C'(t)
0
5
10
15
Time
zoomed
300
250
200
150
100
50
0
C'(t)
0
5
10
15
Time

Fixed α

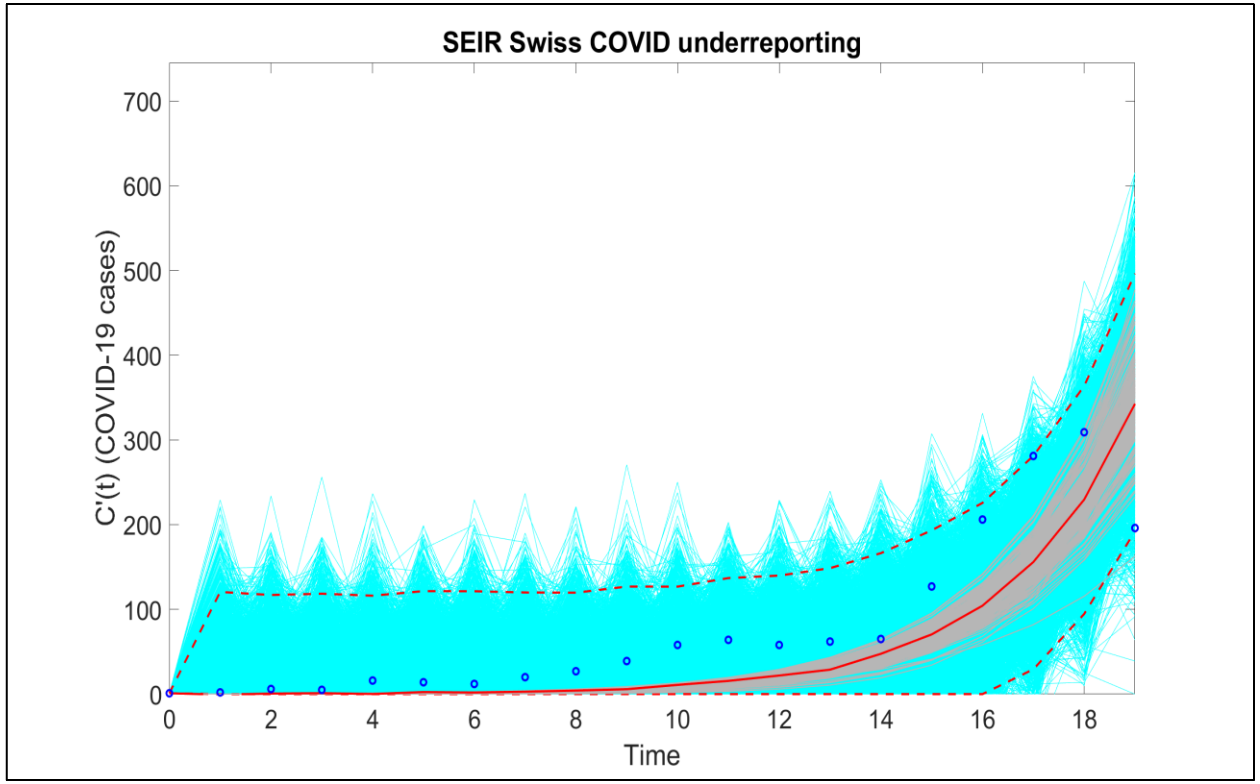


Fixed α

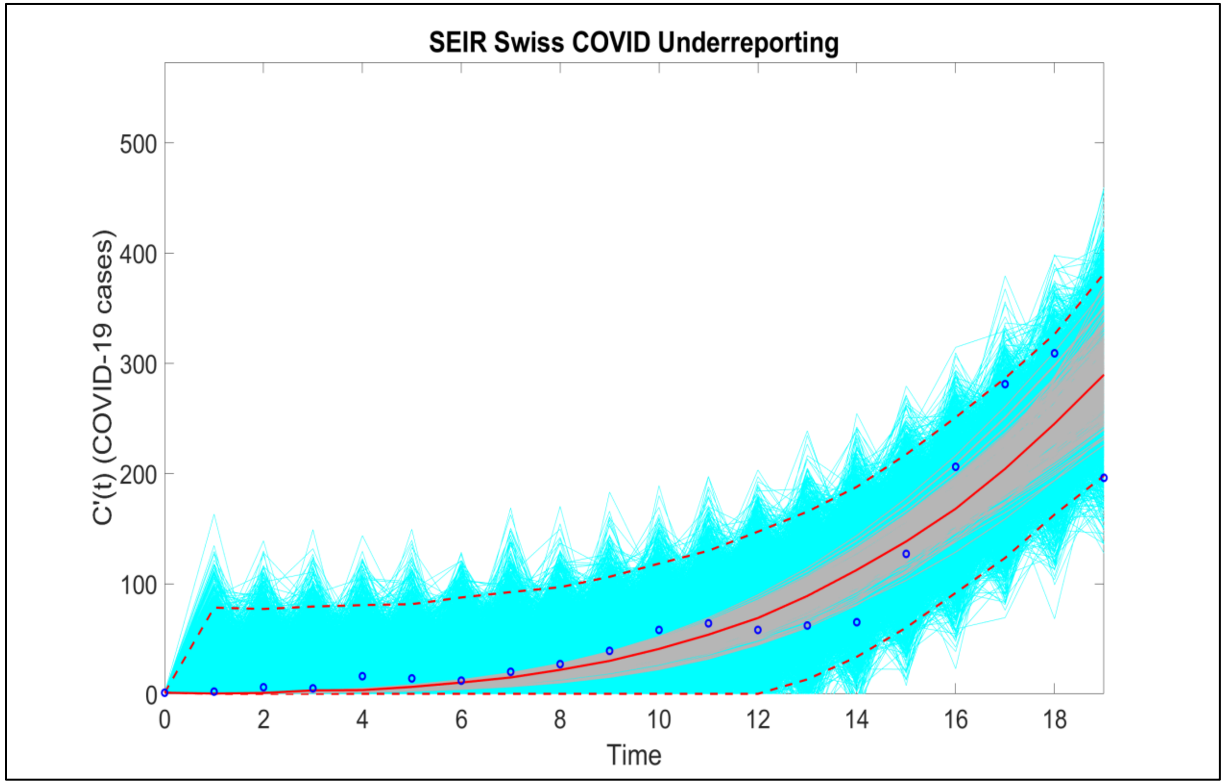
Fitting, plotting, and evaluating the model with quantified uncertainty

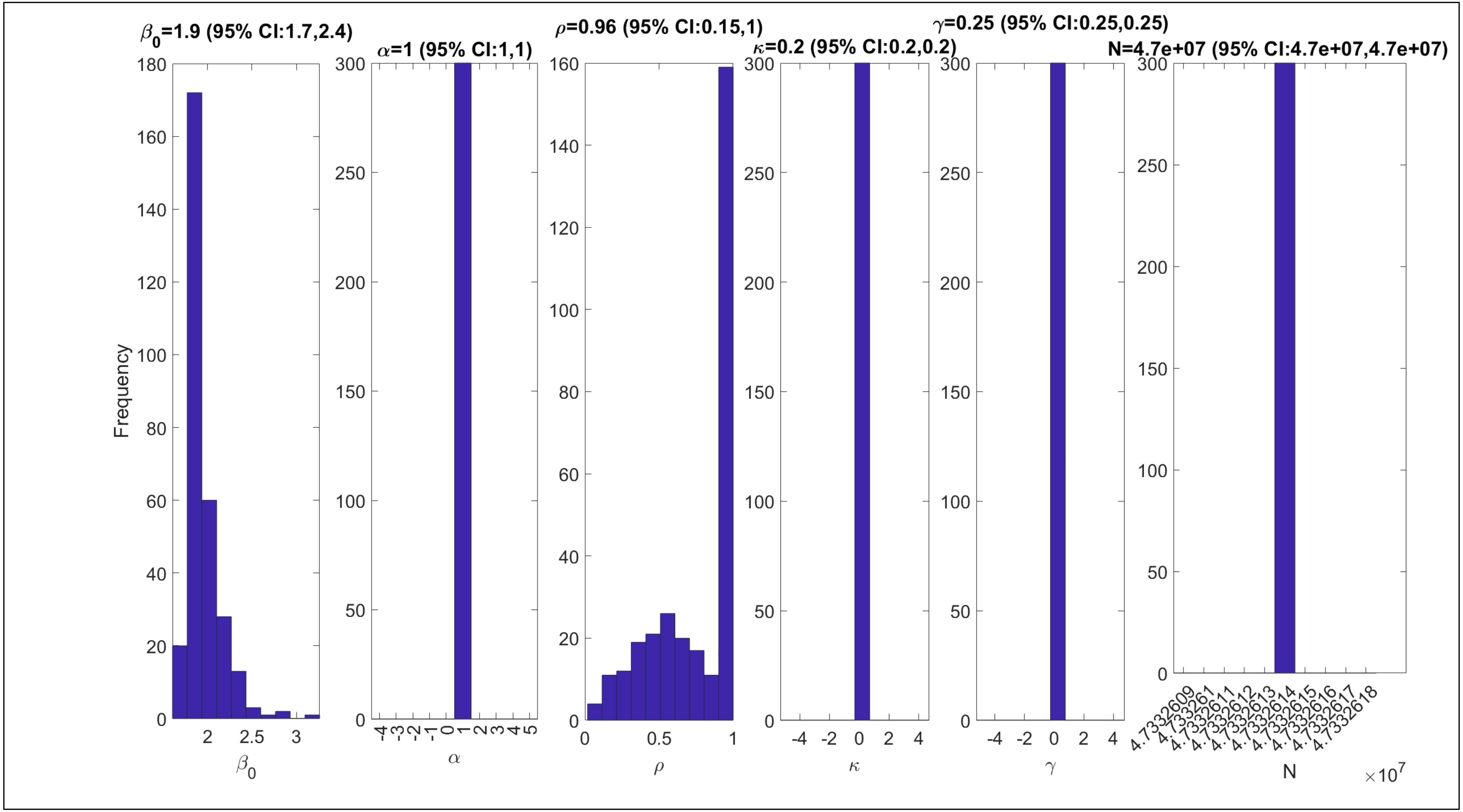
```
(1)Run_Fit_ODEModel(@options_fit_SEIR_unreportedNIn_covid_swiss_dist1_0, 1, 1, 20)
(2)plotFit_ODEModel(@options_fit_SEIR_unreportedNIn_covid_swiss_dist1_0, 1, 1, 20)
```

Fixed α



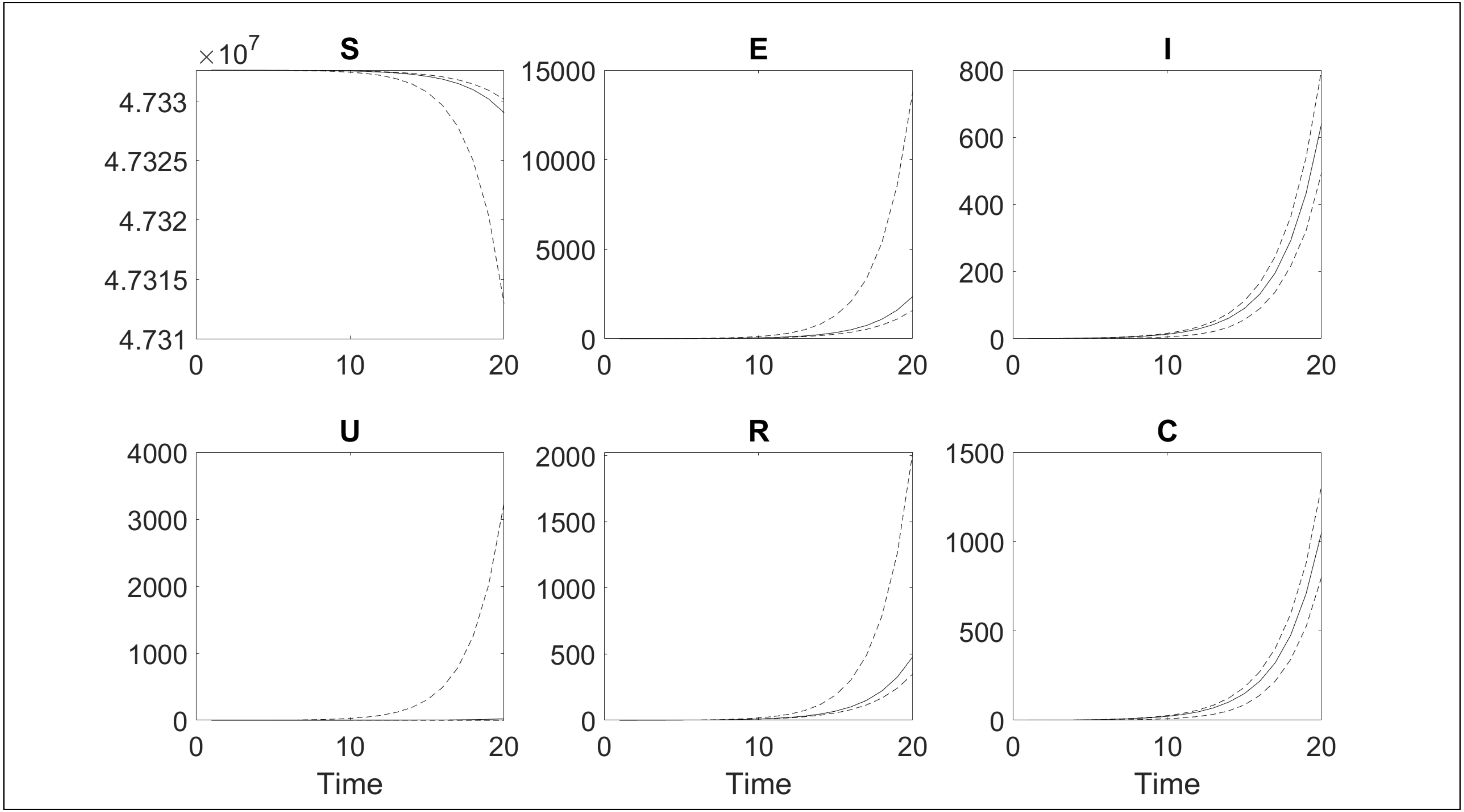
Estimated α





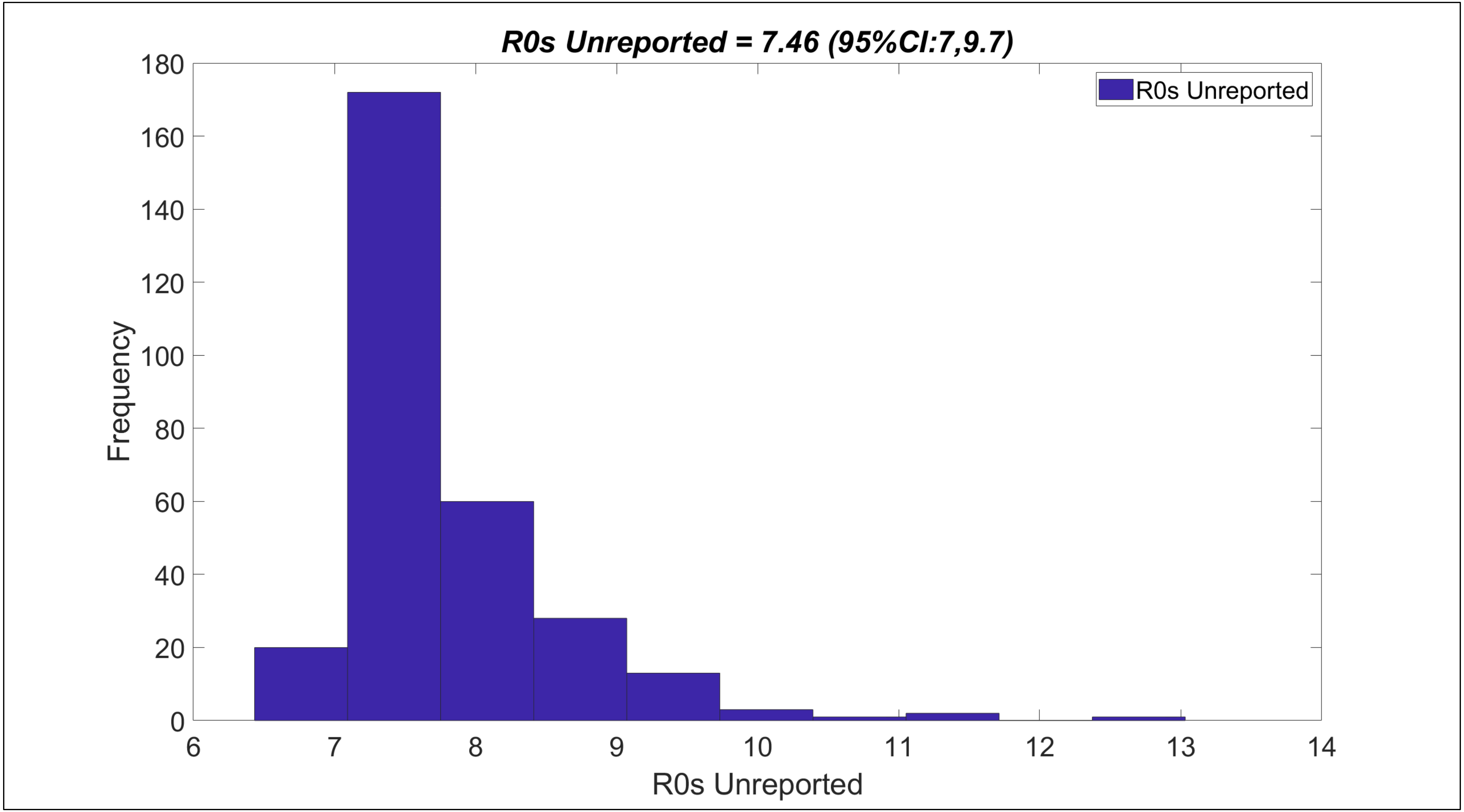
β_0=1.9 (95% CI:1.7,2.4)
α=1 (95% CI:1,1)
ρ=0.96 (95% CI:0.15,1)
κ=0.2 (95% CI:0.2,0.2)
γ=0.25 (95% CI:0.25,0.25)
N=4.7e+07 (95% CI:4.7e+07,4.7e+07)
Frequency
β_0
α
ρ
κ
γ
N
$\times 10^7$

Fixed α



S
$\times 10^7$
4.733
4.7325
4.732
4.7315
4.731
E
15000
10000
5000
I
800
600
400
200
U
4000
3000
2000
1000
R
2000
1500
1000
500
C
1500
1000
500
0
10
20
Time

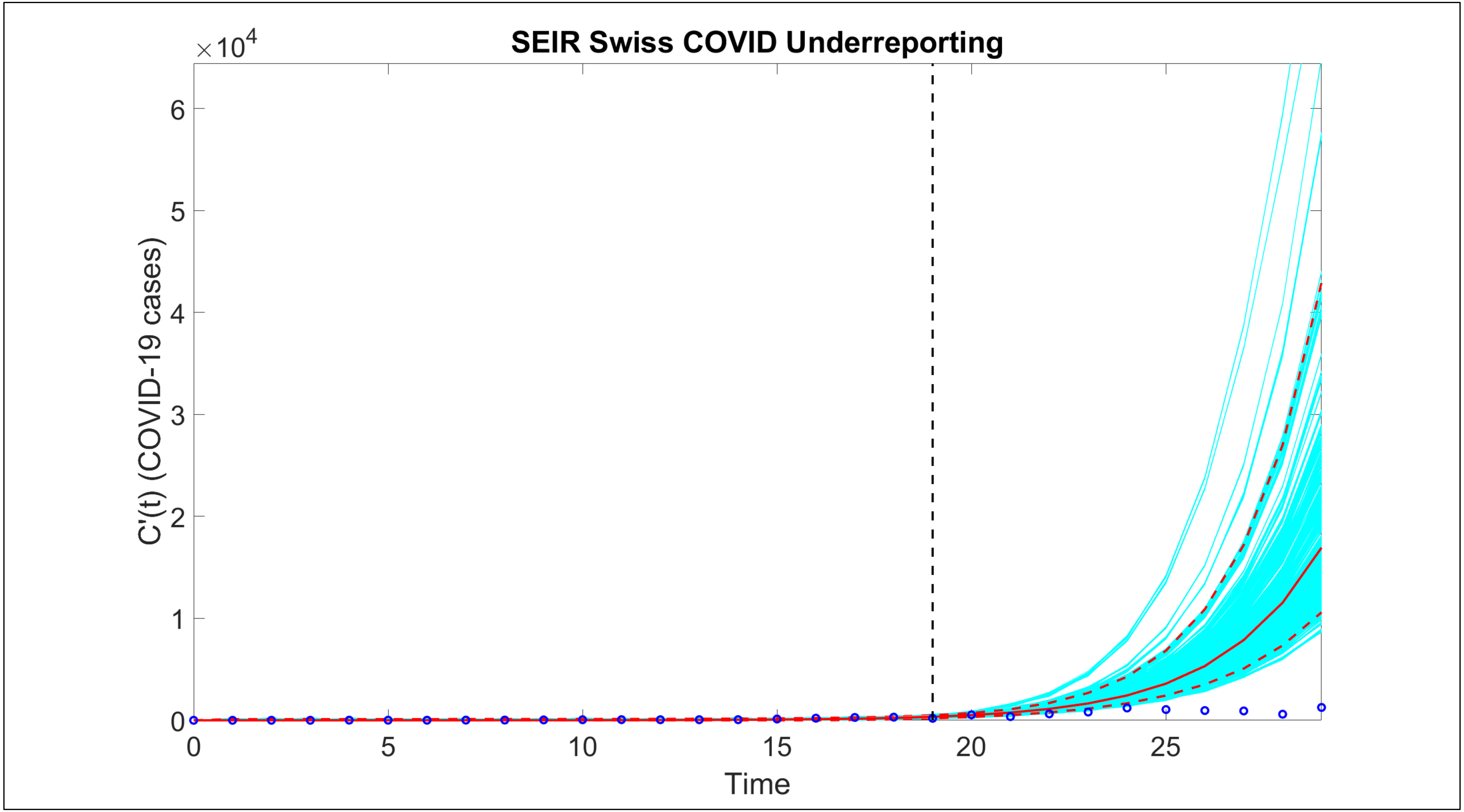
Fixed α


R0s Unreported = 7.46 (95%CI:7,9.7)
R0s Unreported
Frequency
0
20
40
60
80
100
120
140
160
180
6
7
8
9
10
11
12
13
14
R0s Unreported

Fixed α

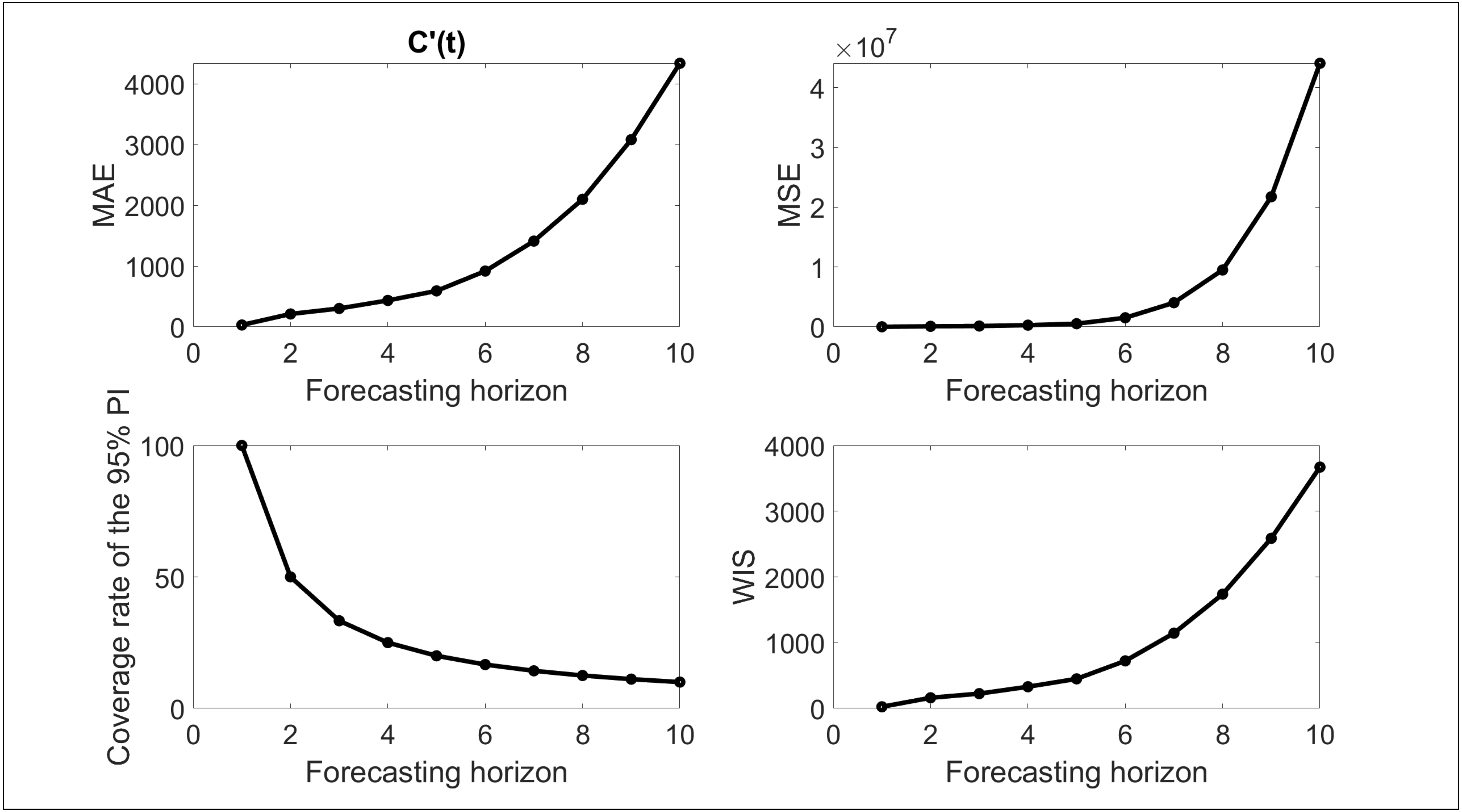
Generating, plotting and assessing model-based forecasts

```
(1)Run_Forecasting_ODEModel(@options_forecast_SEIR_unreportedNIn_covid_swiss_dist1_0, 1, 1, 20,10)
(2)plotForecast_ODEModel(@options_forecast_SEIR_unreportedNIn_covid_swiss_dist1_0, 1, 1, 20,10)
```



SEIR Swiss COVID Underreporting
$\times 10^4$
C'(t) (COVID-19 cases)
Time
0
1
2
3
4
5
6
0
5
10
15
20
25

Fixed α


C'(t)
MAE
4000
3000
2000
1000
0
0
2
4
6
8
10
Forecasting horizon
$\times 10^7$
MSE
4
3
2
1
0
Forecasting horizon
Coverage rate of the 95% PI
100
50
0
Forecasting horizon
WIS
4000
3000
2000
1000
0
Forecasting horizon

Fixed α

Model Comparison (Early Phase)

SEIUR Normal, GGM Normal & RICH Normal

Specifying the GGM Model

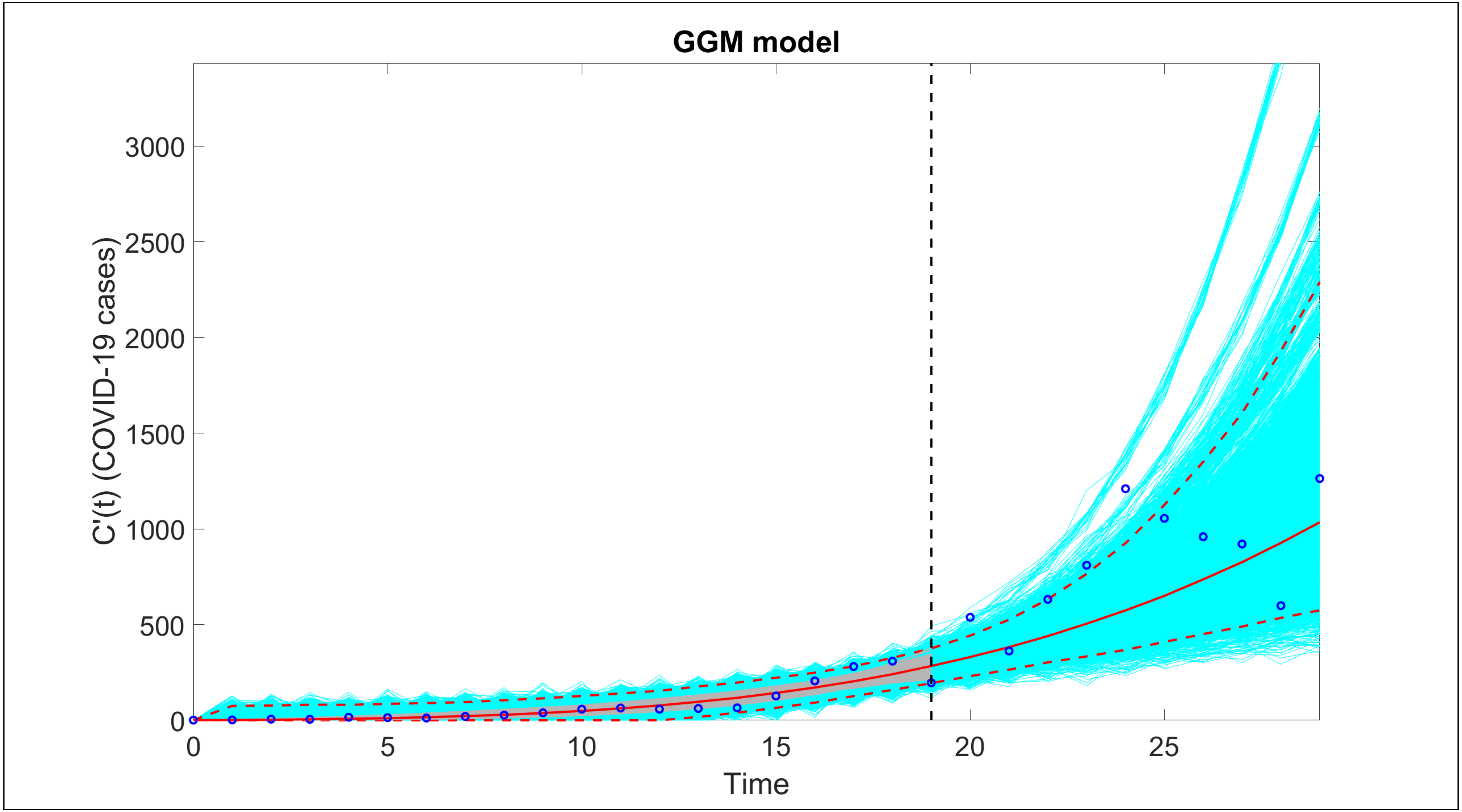
```
% <=============================================================
% < Author: Gerardo Chowell  =====================================
% <=============================================================

function dx=GGM(t,x,params0,extra0)

% parameters in order: r, p

dx=zeros(1,1);

dx(1,1)=params0(1)*x(1,1).^params0(2);
```

$$C'(t) = rC(t)^p$$

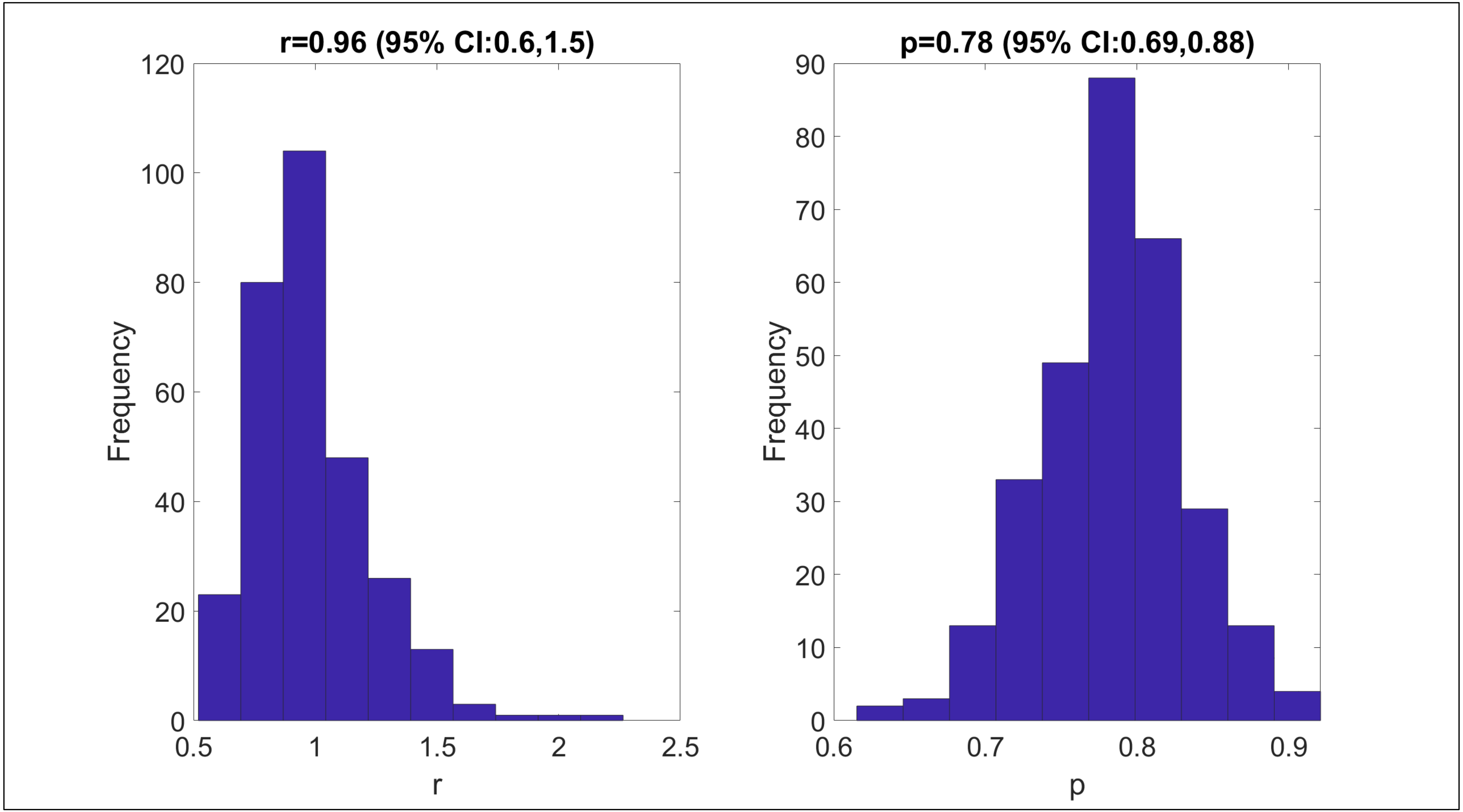
r : Growth rate $(r > 0)$

p : Scaling of growth parameter

```
% <==========================================================================>
% <============================== ODE model ==================================>
% <==========================================================================>

model.fc=@GGM;  % name of the model function
model.name='GGM model'; % string indicating the name of the ODE model

params.label={'r', 'p'}; % list of symbols to refer to the model parameters
params.LB=[0 0];  % lower bound values of the parameter estimates
params.UB=[10 1]; % upper bound values of the parameter estimates
params.initial=[0.91 0.79]; % initial parameter values/guesses
params.fixed=[0 0]; % Boolean vector to indicate any parameters that should remain fixed (1) to initial values indica
params.fixI0=1; % Boolean variable indicating if the initial value of the fitting variable is fixed according to the
params.composite='';  % Estimate a composite function of the individual model parameter estimates otherwise it is lef
params.composite_name=''; % Name of the composite parameter
params.extra0=[]; % used to pass pass any extra parameters (e.g., data, static variables) to the model function

vars.label={'C'}; % list of symbols to refer to the variables included in the model
vars.initial=5; % vector of initial conditions for the model variables
vars.fit_index=1; % index of the model's variable that will be fit to the observed time series data
vars.fit_diff=1; % boolean variable to indicate if the derivative of model's fitting variable should be fit to data.
```



GGM model
C'(t) (COVID-19 cases)
0
500
1000
1500
2000
2500
3000
0
5
10
15
20
25
Time



r=0.96 (95% CI:0.6,1.5)
Frequency
r
p=0.78 (95% CI:0.69,0.88)
Frequency
p

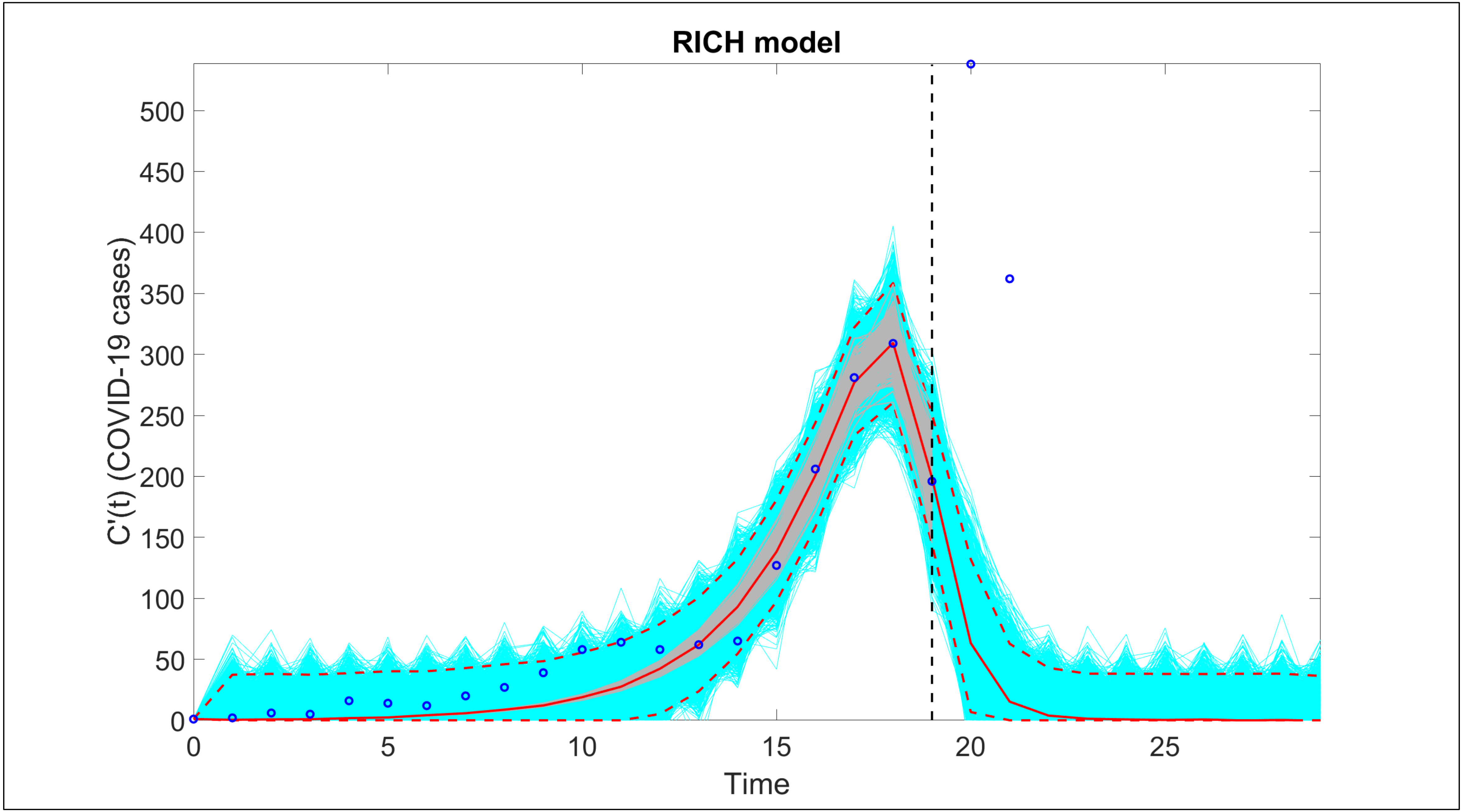
Specifying the RICH Model

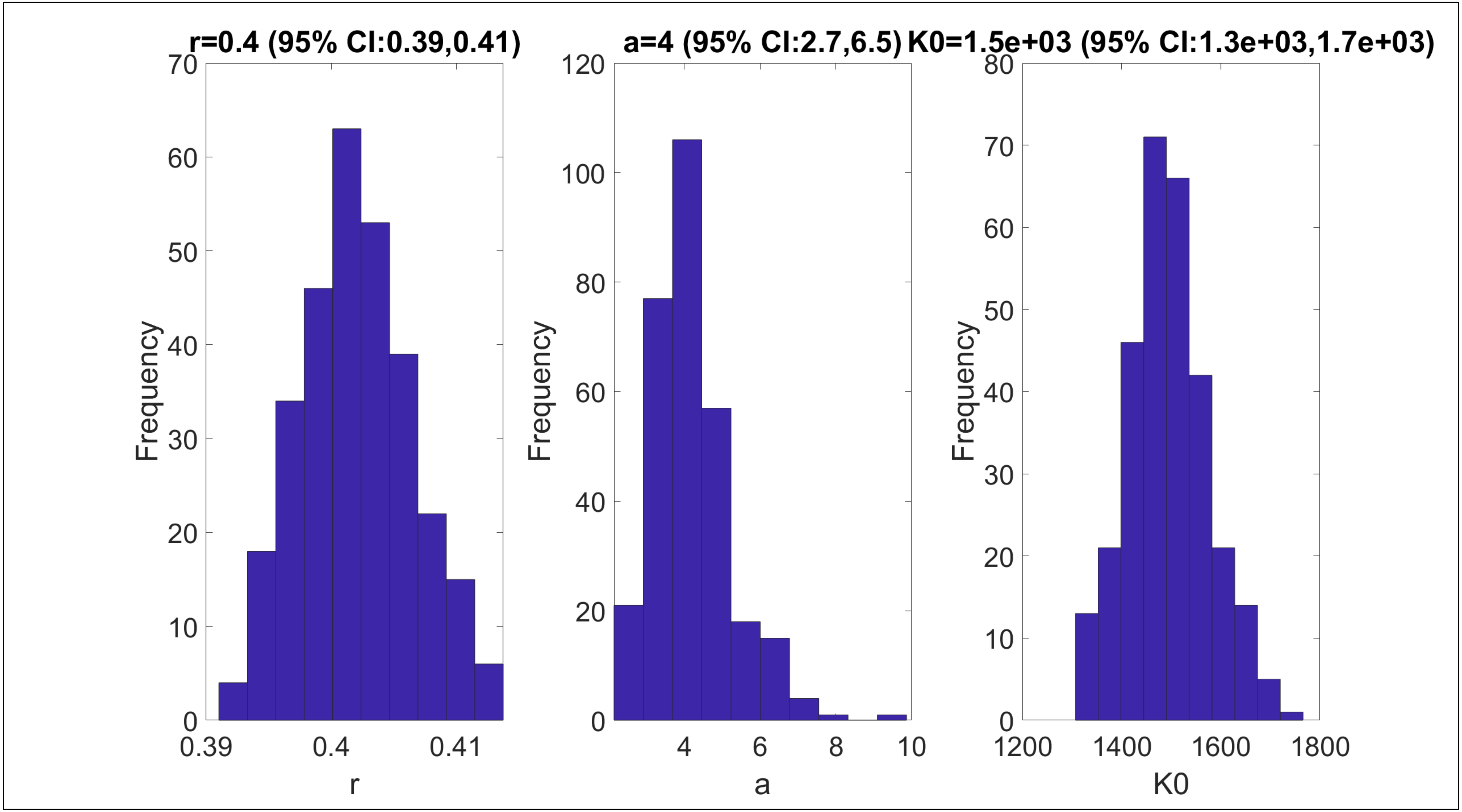
```
% <==================================================================
% < Author: Gerardo Chowell  ===========================================
% <==================================================================

function dx=RICH(t,x,params0,extra0)

% parameters in order: r, a, K0

dx=zeros(1,1);

dx(1,1)=params0(1)*x(1,1)*(1-(x(1,1)/params0(3)).^params0(2));
```

$$C'(t) = rC(t)[1 - \left(\frac{C(t)}{K_0}\right)^a]$$

a: Growth rate $(r > 0)$
p : Scaling of growth parameter
K_0 : Final cumulative epidemic size

```
% <==============================================================================>
% <============================== ODE model =====================================>
% <==============================================================================>

model.fc=@RICH;  % name of the model function
model.name='RICH model'; % string indicating the name of the ODE model

params.label={'r', 'a', 'K0'}; % list of symbols to refer to the model parameters
params.LB=[0 0 10];  % lower bound values of the parameter estimates
params.UB=[10 10 40000]; % upper bound values of the parameter estimates
params.initial=[0.91 0.79 30994]; % initial parameter values/guesses
params.fixed=[0 0 0]; % Boolean vector to indicate any parameters that should remain fixed (1) to initial values indi
params.fixI0=1; % Boolean variable indicating if the initial value of the fitting variable is fixed according to the
params.composite='';  % Estimate a composite function of the individual model parameter estimates otherwise it is let
params.composite_name=''; % Name of the composite parameter
params.extra0=[]; % used to pass pass any extra parameters (e.g., data, static variables) to the model function

vars.label={'C'}; % list of symbols to refer to the variables included in the model
vars.initial=5; % vector of initial conditions for the model variables
vars.fit_index=1; % index of the model's variable that will be fit to the observed time series data
vars.fit_diff=1; % boolean variable to indicate if the derivative of model's fitting variable should be fit to data.
```


RICH model
C'(t) (COVID-19 cases)
0
50
100
150
200
250
300
350
400
450
500
0
5
10
15
20
25
Time


r=0.4 (95% CI:0.39,0.41)
a=4 (95% CI:2.7,6.5)
K0=1.5e+03 (95% CI:1.3e+03,1.7e+03)
Frequency
r
a
K0
0.39
0.4
0.41
4
6
8
10
1200
1400
1600
1800

Model	MAE	MSE	Coverage 95% PI	WIS
Calibration Performance				
SEIUR model with NLSQ/ Normal error structure (<dist1>=0)	40.1	3299.6	**90.0**	23.8
GGM with Normal error structure (<dist1>=0)	**21.1**	1191.0	**100.0**	15.2
RICH with negative binomial error structure (<dist1>=0)	22.6	**1037.1**	**100.0**	**14.0**
Forecast Performance				
SEIUR model with NB error structure (<dist1>=0)	4475.0	46849766.5	10.0	3893.0
GGM with negative binomial error structure (<dist1>=0)	**254.6**	**91255.1**	**70.0**	**171.5**
RICH with negative binomial error structure (<dist1>=0)	533.4	368516.5	20.0	428.6

Specifying `SEIURC`

All Data

```
% <=========================================================================>
% < Author: Gerardo Chowell  ==================================================>
% <=========================================================================>

function dx=SEIR1(t,x,params0,extra0)

beta0=params0(1);
beta1=params0(2);
alpha=params0(3); % non-homogenous mixing parameter
q1=params0(4);
rho=params0(5);
k=params0(6);
gamma1=params0(7);
N=params0(8);

if t<20

    betaf=beta0;

else

    %betaf=beta0*(1./(t+1)).^q1; % power-law decline

    %betaf=beta0*exp(-q1*t); % exponential decline

    betaf=beta1+(beta0-beta1)*exp(-q1*(t-20)); % exponential decline

end

dx=zeros(6,1);  % define the vector of the state derivatives: S, E, I, U, R, C

dx(1,1)= -betaf*x(1,1).*((x(3,1)+x(4,1)).^alpha)./N; %S

dx(2,1)= betaf*x(1,1).*((x(3,1)+x(4,1)).^alpha)./N - k*x(2,1); %E

dx(3,1)= k*rho*x(2,1) - gamma1*x(3,1); %I

dx(4,1)= k*(1-rho)*x(2,1) - gamma1*x(4,1); %U

dx(5,1)= gamma1*(x(3,1)+x(4,1)); %R

dx(6,1)= k*rho*x(2,1); %C
```

$$f(\beta) = \begin{cases} \beta_0, & t < 20 \\ \beta_1 + (\beta_0 - \beta_1)e^{-q1(t-20)}, & Otherwise \end{cases}$$

$$\begin{cases} \dot{S} = -f(\beta)S(t)\dfrac{I(t) + U(t)^{\alpha}}{N} \\ \dot{E} = f(\beta)S(t)\dfrac{I(t) + U(t)^{\alpha}}{N} - \kappa E(t) \\ \dot{I} = \kappa\rho E(t) - \gamma I(t) \\ \dot{U} = \kappa(1-\rho)E(t) - \gamma U(t) \\ \dot{R} = \gamma(I(t) + U(t)) \\ \dot{C} = \kappa\rho E(t) \end{cases}$$

```
% <=========================================================================>
% <============================== ODE model ================================>
% <=========================================================================>

model.fc=@SEIR_unreported_expo_decline; % name of the model function
model.name='SEIR swiss_covid underreporting';   % string indicating the name of the ODE model

params.label={'\beta_0','\beta_1','\alpha','q','\rho','\kappa','\gamma','N'};  % list of symbols to refer to the model parameters
params.LB=[0.001 0AF 0.5 0 0.01 0.01 0.01 47332614]; % lower bound values of the parameter estimates
params.UB=[5 2 1 10 1 1 1 47332614]; % upper bound values of the parameter estimates
params.initial=[0.6 0.01 1 0.01 1 1/5 1/4 47332614]; % initial parameter values/guesses
params.fixed=[0 0 1 0 0 1 1 1]; % Boolean vector to indicate any parameters that should remain fixed (1) to initial values indicated in params.
params.fixI0=1; % Boolean variable indicating if the initial value of the fitting variable is fixed according to the first observation in the t
params.composite=@R0s_unreported;  % Estimate a composite function of the individual model parameter estimates otherwise it is left empty.
params.composite_name='R0s_unreported'; % Name of the composite parameter
params.extra0='';

vars.label={'S','E','I','U','R','C'}; % list of symbols to refer to the variables included in the model
vars.initial=[params.initial(8)-1 0 1 0 0 1];  % vector of initial conditions for the model variables
vars.fit_index=6; % index of the model's variable that will be fit to the observed time series data
vars.fit_diff=1; % boolean variable to indicate if the derivative of model's fitting variable should be fit to data.
```

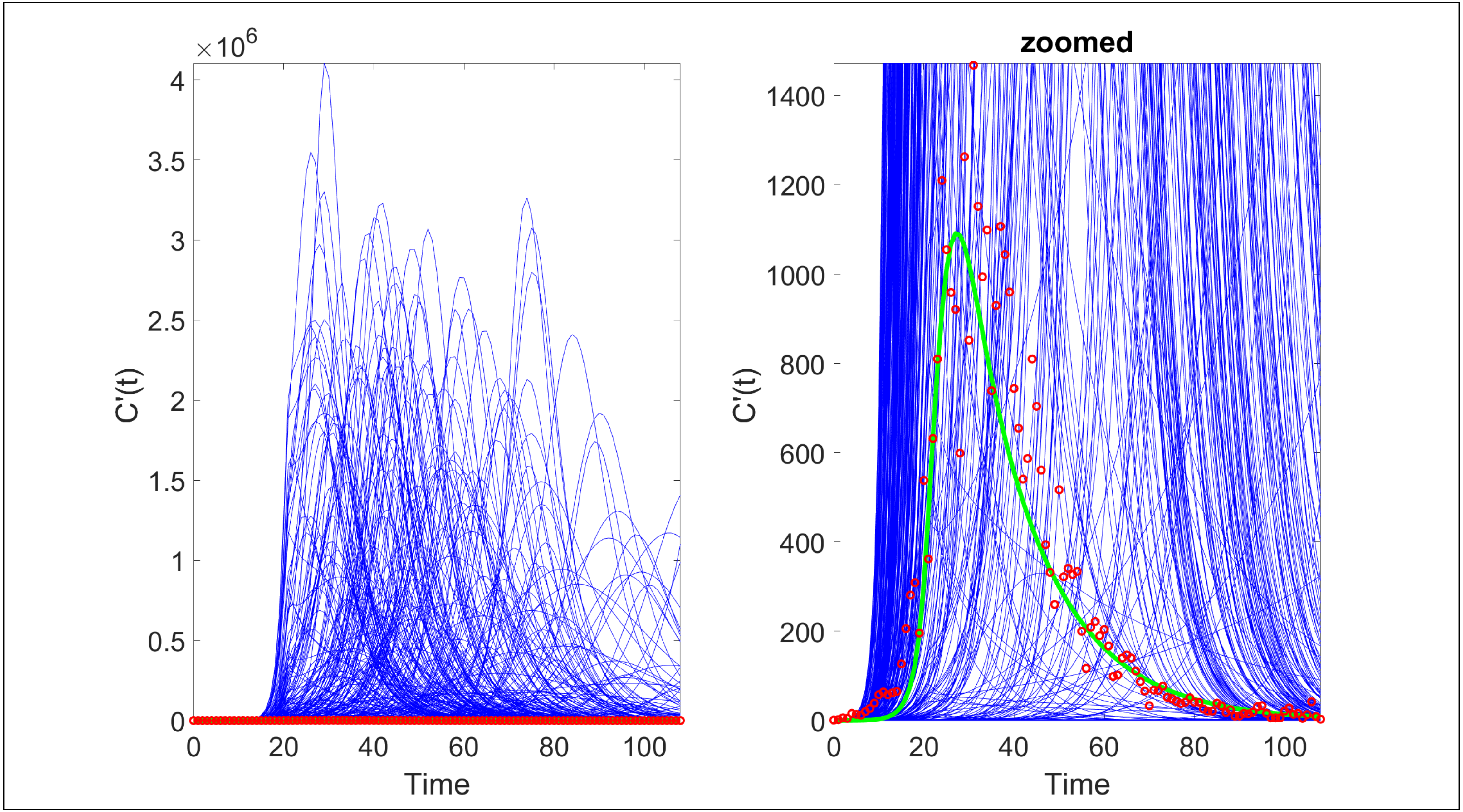
Preparing `options_fit.m`

Fitting using all data

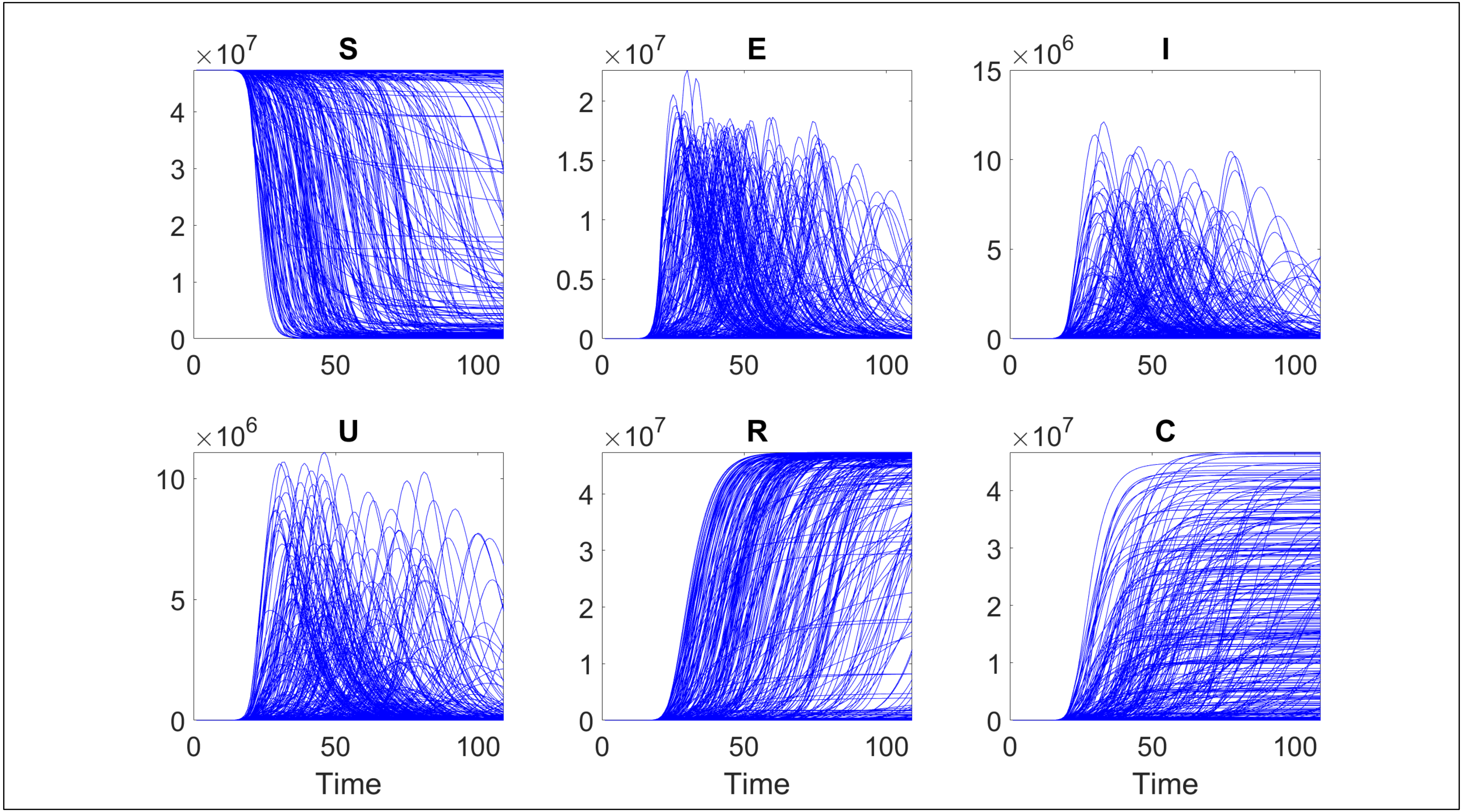
```
% <==================================================================================>
% <========================== Parameters of the rolling window analysis =========================>
% <==================================================================================>

windowsize1=109;  % moving window size

tstart1=1; % time point for the start of rolling window analysis

tend1=1;  %time point for the end of the rolling window analysis

printscreen1=1;
```

Generating preliminary model solutions

```
plotODEModel(@options_fit_SEIR_unreported_covid_swiss_dist1_0)
```



Estimated α

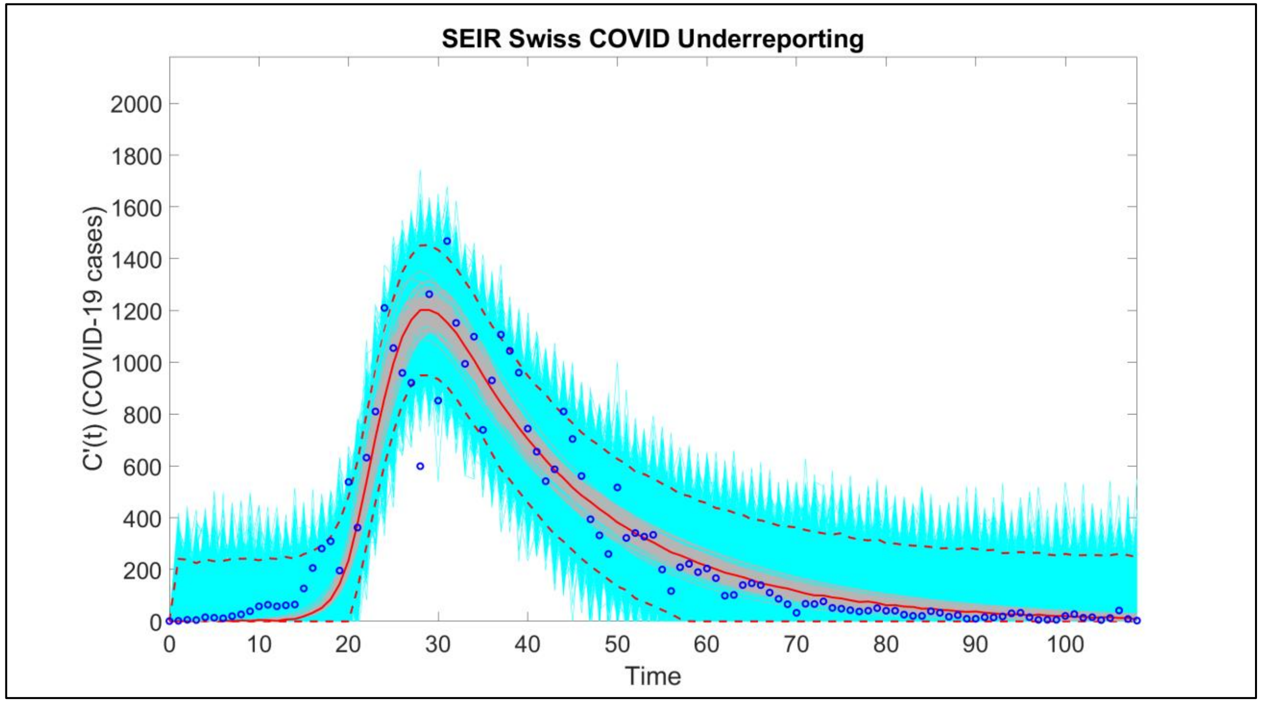


Estimated α

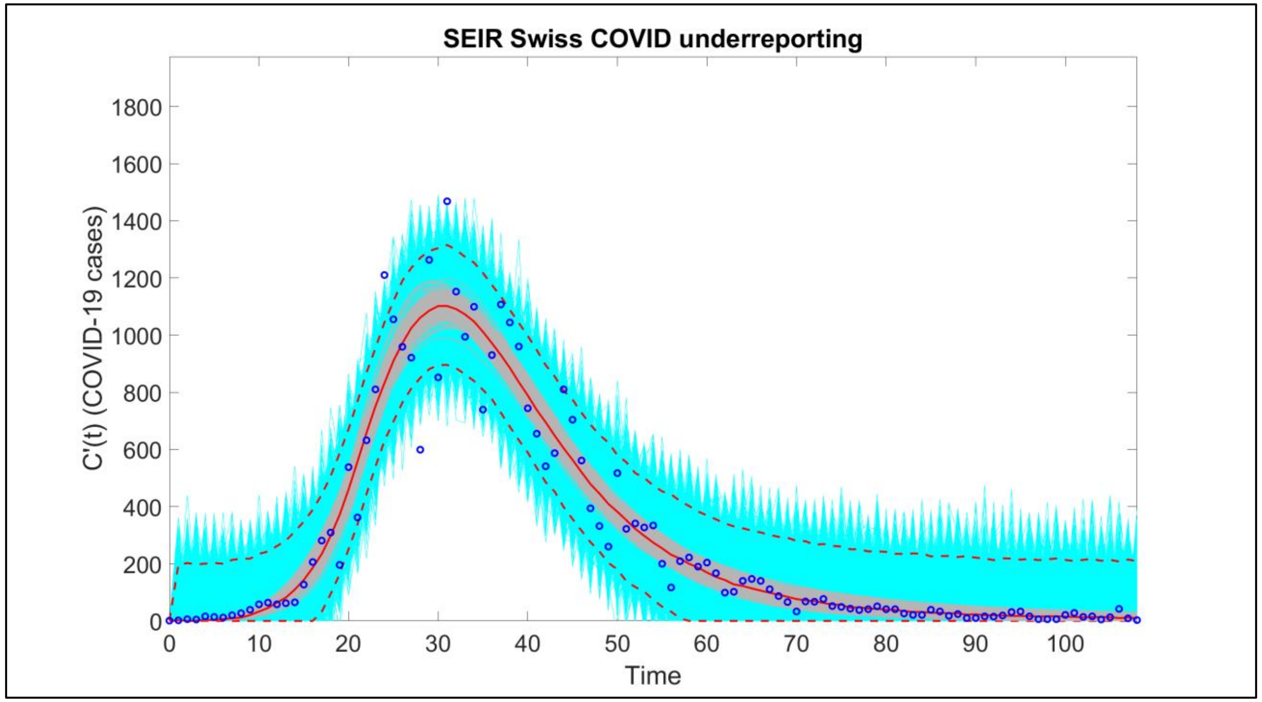
Fitting, plotting, and evaluating the model with quantified uncertainty

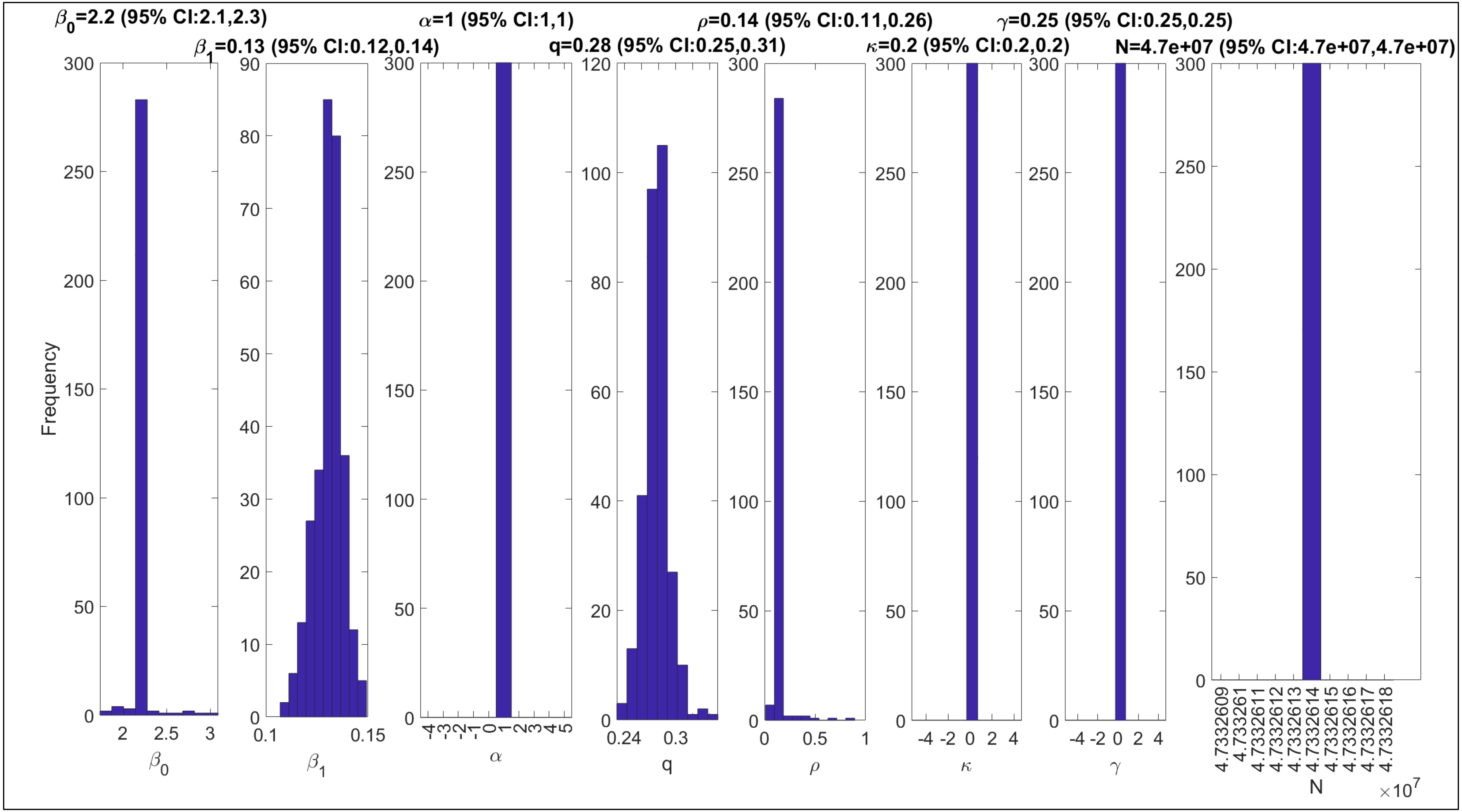
```
(1)Run_Fit_ODEModel(@options_fit_SEIR_unreported_covid_swiss_dist1_0, 1, 1, 109)
(2)plotForecast_ODEModel(@options_fit_SEIR_unreported_covid_swiss_dist1_0, 1, 1, 109)
```

Fixed α

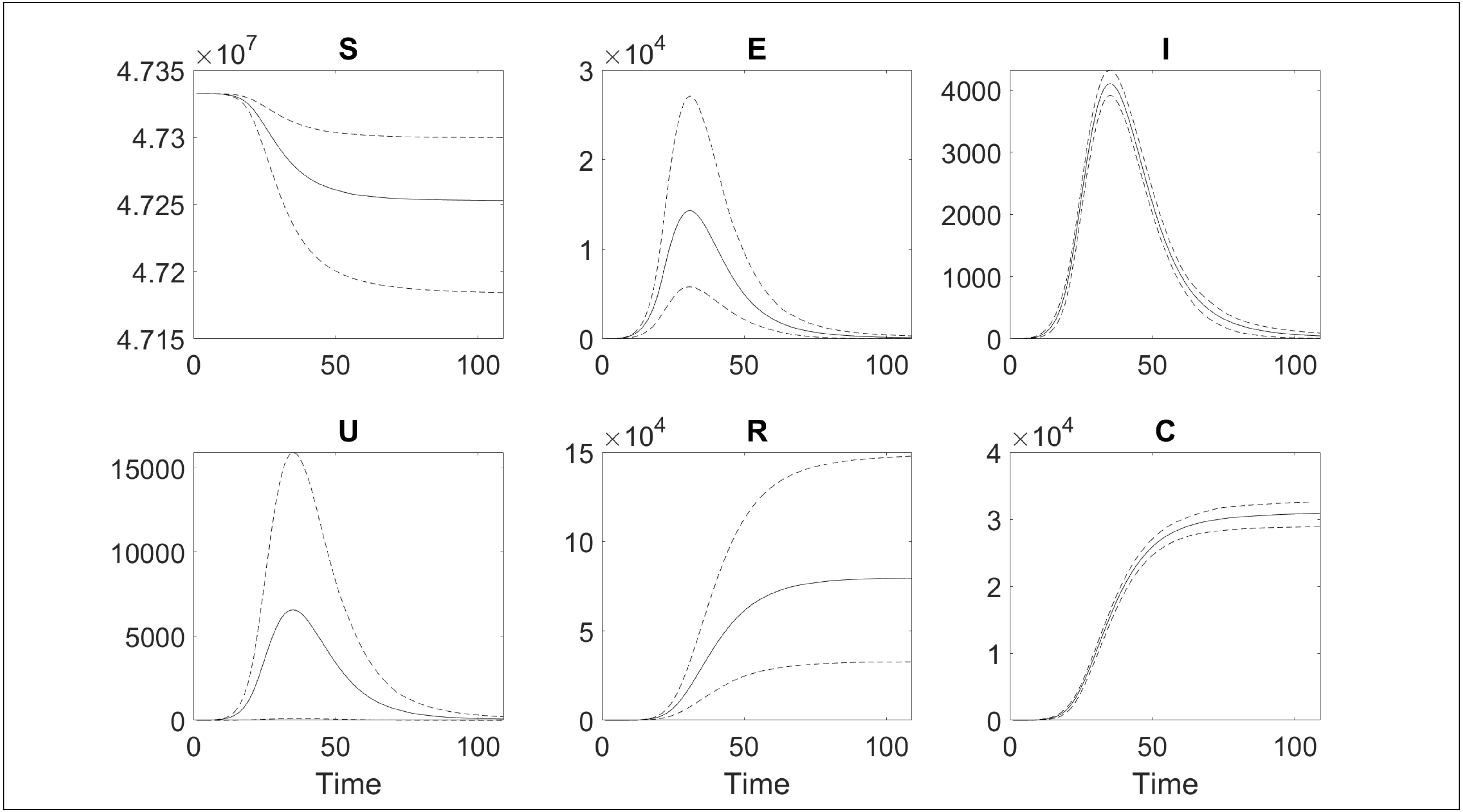


Estimated α





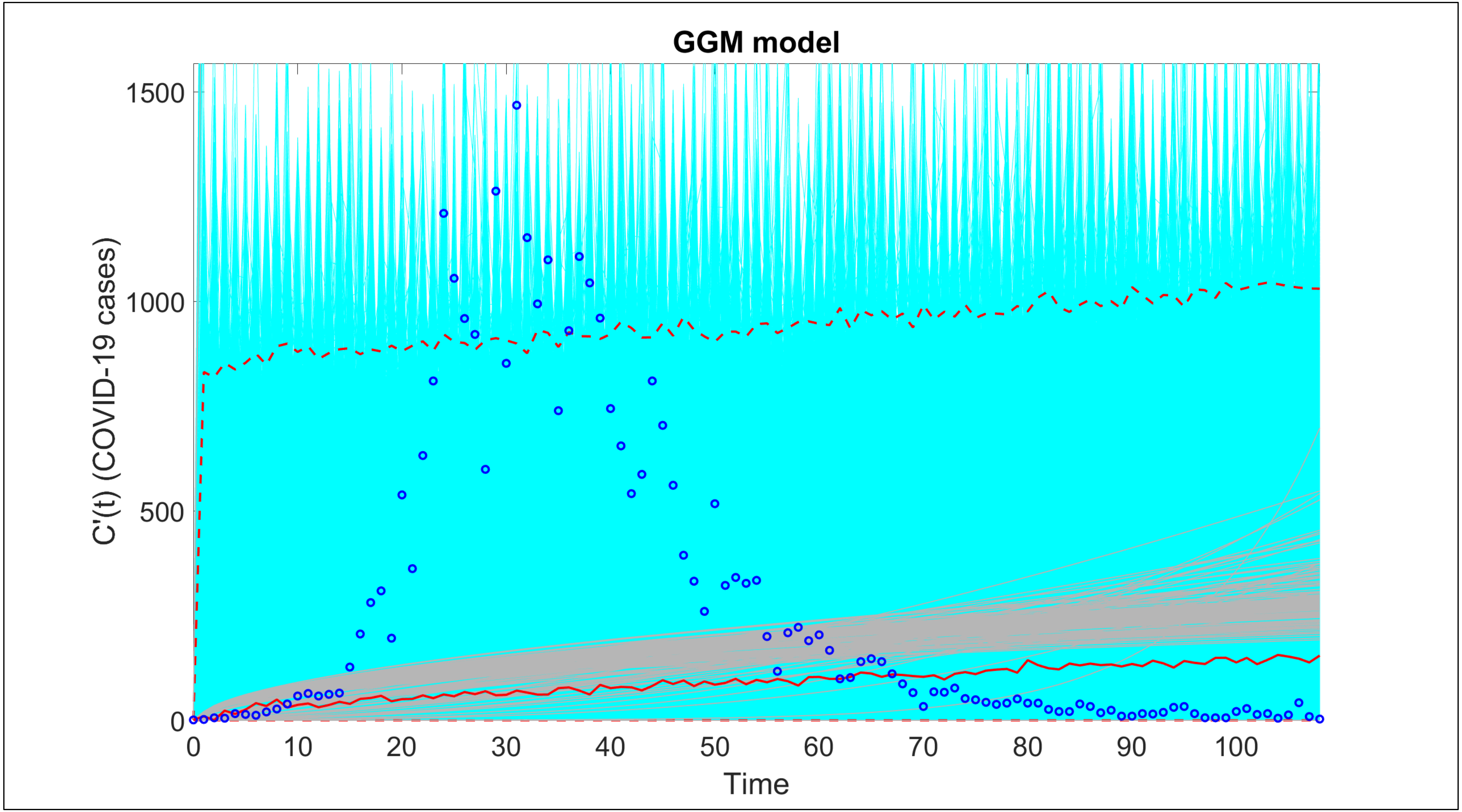
Estimated α

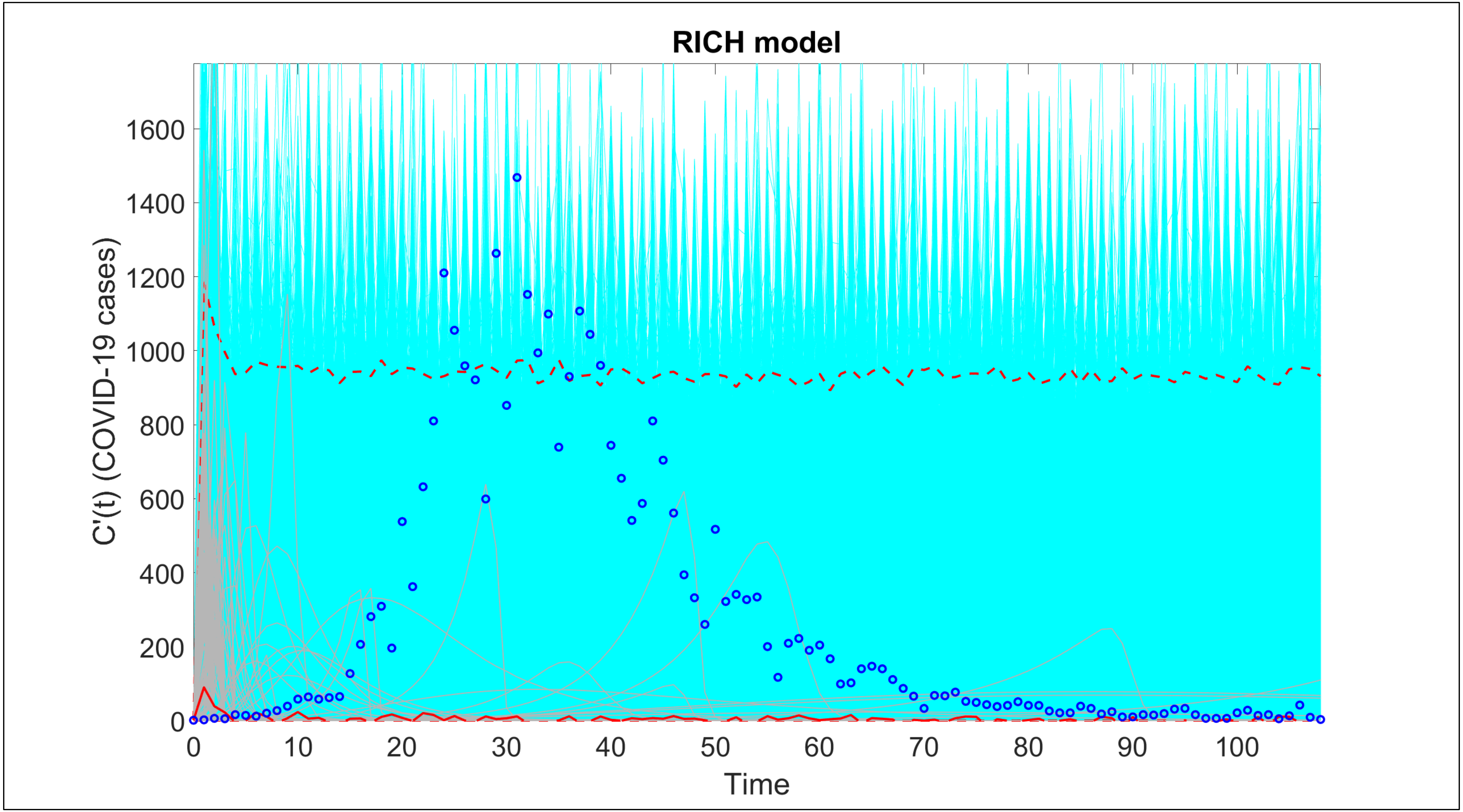


Estimated α

Model Comparison (All Data)

SEIUR Normal, GGM Normal & RICH Normal


GGM model
C'(t) (COVID-19 cases)
1500
1000
500
0
0
10
20
30
40
50
60
70
80
90
100
Time


RICH model
C'(t) (COVID-19 cases)
0
200
400
600
800
1000
1200
1400
1600
0
10
20
30
40
50
60
70
80
90
100
Time

Model	MAE	MSE	Coverage 95% PI	WIS
Calibration Performance				
SEIUR model with NLSQ/ Normal error structure (<dist1>=0)	**67.2**	**13189.4**	**92.7**	**48.3**
GGM with Normal error structure (<dist1>=0)	297.6	197605.8	88.1	176.6
RICH with negative binomial error structure (<dist1>=0)	284.3	216949.7	89.0	173.1