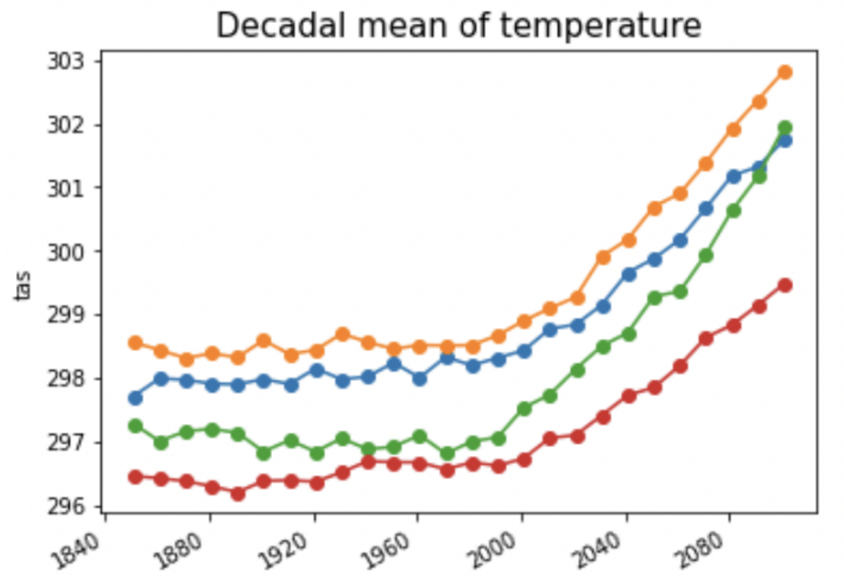
Homework 6

1.Climate data analysis

Answer 1(a)

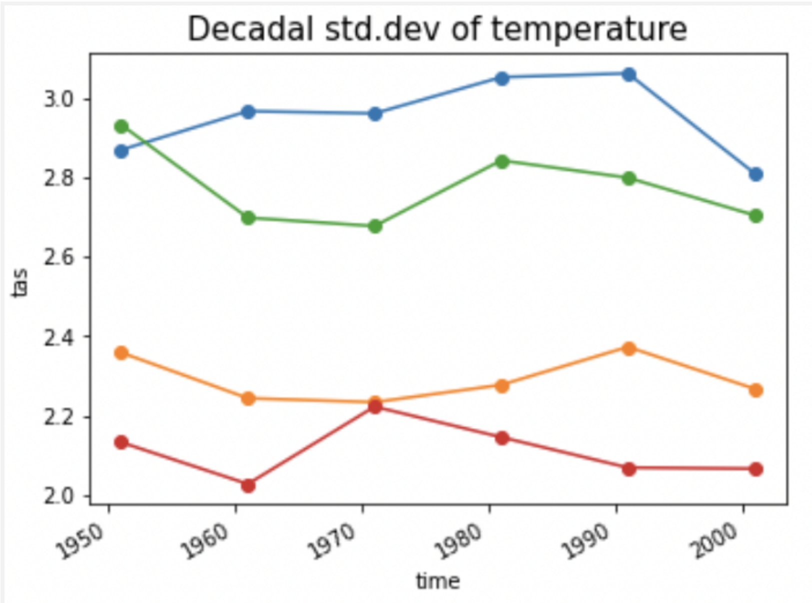


Comments:

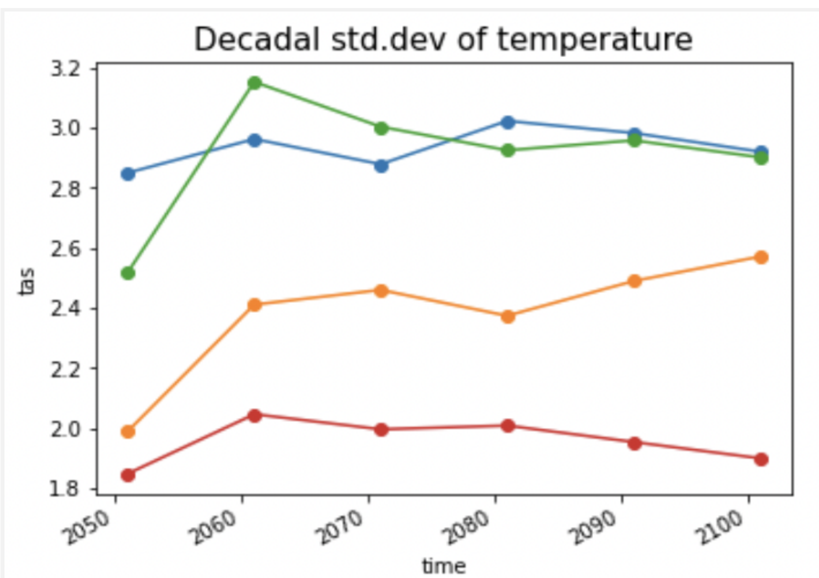
The qualitative features of the plot is the surface air temperature plotted on the y-axis. To further point out another qualitative feature- There is seen to be a rise in mean temperatures since 2000 because of industrialization and population.

For all the 4 models the decadal mean temperature is seen to be increasing but slight difference because there might be some error as it spans over long periods of time.

Answer 1(b)
Between 1950-2000



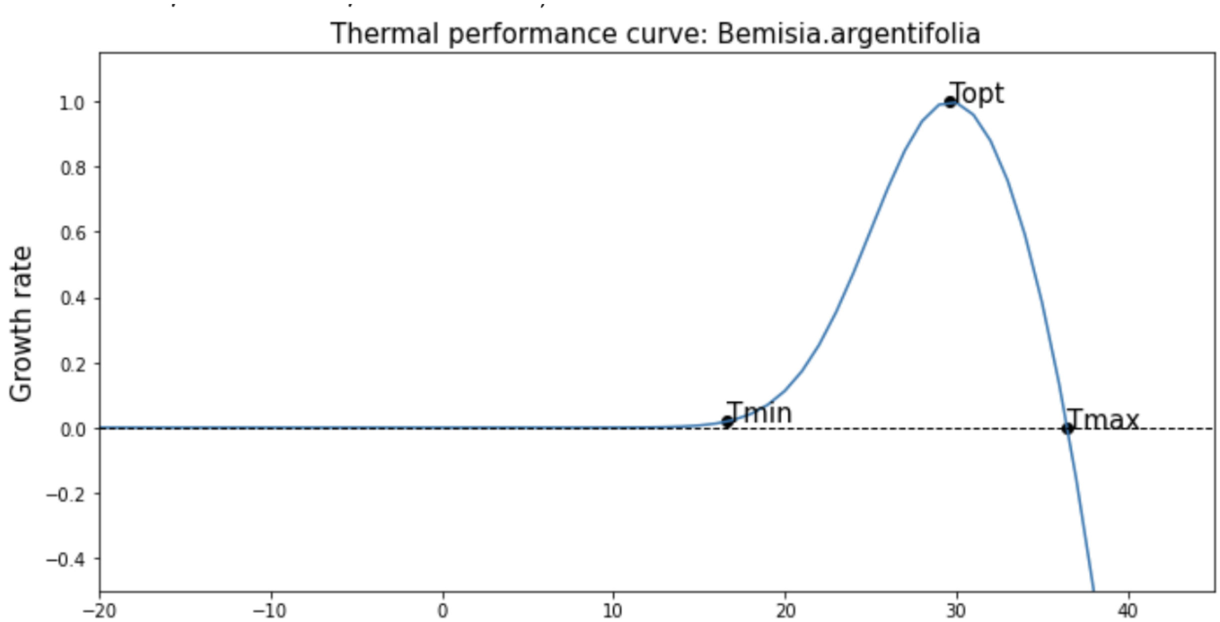
Between 2050-2100



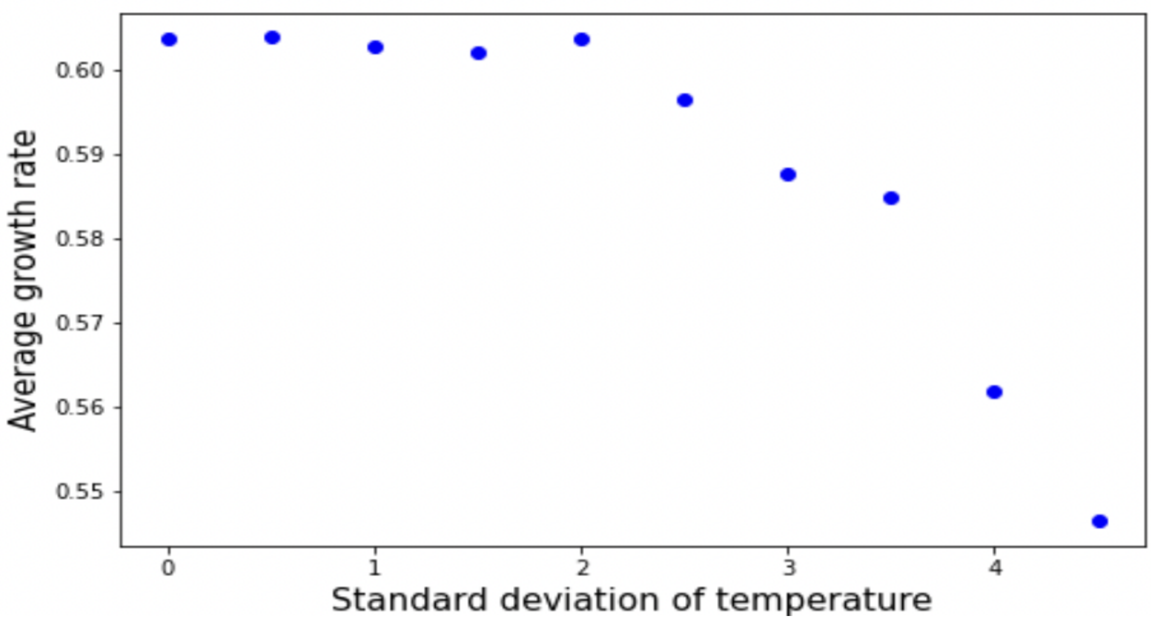
In the range of 1950-2000: For the 3 models depicted with the red, blue and Green trend lines in the plot are seen to have statistically significant changes.
In the range of 2050-2100: For the 2 models depicted with green and orange trend lines in the plot are seen to have statistically significant changes.

2)**Ecology modeling**:

Answer 2(a)



Answer 2(b)

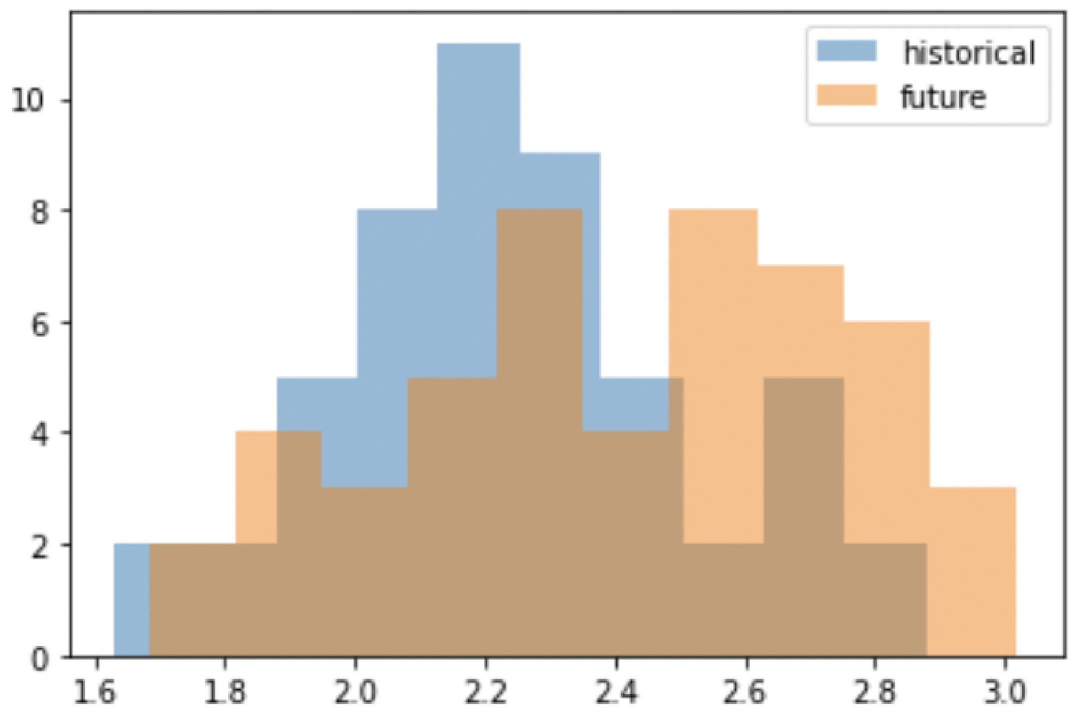


Comments:

The plot shows how growth rate is seen to vary depending on the variation of temperature. As the std. dev of the temperature increases, the growth rate i.e performance of the species given starts to decrease overall.

Answer 2(c):

Selected model_2 for this section



Output:

```
t(df=99) = -2.61, p = 0.0106
Samples do not have identical means (reject H0)

This implies:
```

1. Degrees of freedom: 99
2. Test statistic at 99 degrees of freedom= -2.61
3. P- value = 0.0106

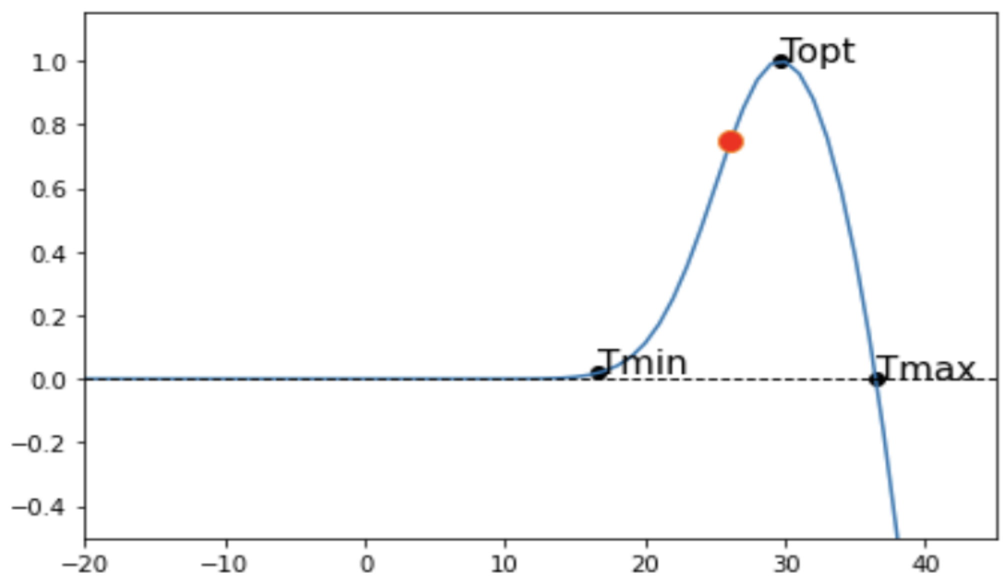
Reject the null hypothesis implying that we accept the alternate hypothesis that there is a statistically significant difference between model_2 for historical and future periods.

Answer 2(d)

`Mean temperature in Celcius for Bemisia argentifolia is:array(26.96113302)`

Output shows that The mean temperature is = 26 degrees Celcius.
This mean temperature falls close to the optimum temperature of 30 deg celsius.
It is far away from the maximum and minimum temperatures.

The red data point in the plot below VISUALIZES where the mean temperature falls wrt. To min, max and optimal temperature.



BONUS:

The temperature variability won't be beneficial to the growth rate at this temperature. Reason: The second derivative graph, shown below, shows that temperature of 26C (orange data point) falls on the negative spectrum and we know that only temperature variations in the positive spectrum here are beneficial)

3. Literature

(i) Why Bio-diversity loss is a global risk and how climate change is impacting loss of biodiversity and ecology specifically ectotherms

The majority of models predict dire effects for biodiversity, with the worst-case scenarios resulting in rates of extinction comparable to the sixth mass extinction in Earth's history.Thermal tolerance data for a variety of vertebrate taxa show comparable trends, implying that these findings are applicable to all terrestrial ectotherms. Findings suggest that, in the absence of mitigating variables like migration and adaptation, the largest extinction risks from global warming may be found in the tropics, which also have the most biological variety.Individuals, populations, species, ecological networks, and ecosystems are all affected by varying strengths and forms of fitness loss, which manifest themselves at different levels and affect individuals, populations, species, ecological networks, and ecosystems. Climate change has the potential to reduce genetic variety of populations at the most basic levels of biodiversity due to directional selection and fast migration, which could have an impact on ecosystem functioning and resilience.

(ii)Trends in temporal autocorrelation and scale- specific variability in the world over last 100 years

Temperature autocorrelation can have a major impact on community structure and durability by increasing the duration and severity of unfavorable conditions in sink populations and disrupting spatial rescue effects by synchronizing geographically dispersed populations. Although historical data has demonstrated an increase in temperature spatial and temporal autocorrelation, nothing is known about how climate change can alter these patterns. We used daily air temperature data from 21 General Circulation Models under the business-as-usual carbon emission scenario to assess patterns of spatial and temporal autocorrelation between 1871 and 2099. Although both spatial and temporal autocorrelation increased over time, there was significant regional variation in temporal autocorrelation trends. Furthermore, a consistent discontinuity in the relationship between spatial autocorrelation time and spatial autocorrelation time was identified.

(iii) Why cross sensor learning with Low Earth Orbit (LEO) and Geostationary (GEO) sensor approaches works well for estimation of Land Surface Temperatures (LST)

Remote sensing from space provides continuous, area-wide data on the state of the Earth-atmosphere system and its components. Various satellite platforms with various sensors have been built over the last 50 years to monitor atmospheric parameters used in meteorological and climatological studies, and the data retrieved from satellite-based sensors has greatly improved our understanding of the processes and dynamics within the Earth-atmosphere system. The current paper gives an overview of existing satellites and sensors, as well as known methods for obtaining meteorological and climatological parameters. It also gives information about new systems that are being developed in the near future.Although traditional studies have largely focused on determining the ecological consequences of changing average environmental conditions, there is growing recognition that statistical properties beyond the mean can play an equally important role in shaping the structure and functioning of ecosystems in the face of climate change.

```
import cftime
import matplotlib.pyplot as plt
import numpy as np
import os
import pandas as pd
from scipy import stats
import scipy.integrate as inte
from scipy.stats import shapiro
import xarray as xr



# Using R inside python
import rpy2
import rpy2.robjects.packages as rpackages
from rpy2.robjects.packages import importr
utils = rpackages.importr('utils')
utils.chooseCRANmirror(ind=1)
utils.install_packages('synchrony')
synchrony = importr('synchrony')
rstats = importr('stats')
```

```
R[write to console]: Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

R[write to console]: trying URL 'https://cloud.r-project.org/src/contrib/synchro

R[write to console]: Content type 'application/x-gzip'
R[write to console]:  length 34687 bytes (33 KB)

R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
```

```
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
```

Downloading CMIP6 data (not necessary for this tutorial)

CMIP6 search interface: https://esgf-node.llnl.gov/search/cmip6/

Import data from Github

```
!wget https://github.com/KateDuffy/climate-ecology-tutorial/blob/main/tas_day_BCC-CSM2
os.rename('tas_day_BCC-CSM2-MR_historical+ssp585_1850-2100_Bemisia.argentifolia.nc?raw

!wget  https://github.com/KateDuffy/climate-ecology-tutorial/blob/main/tas_day_CESM2_h
os.rename('tas_day_CESM2_historical+ssp585_1850-2100_Bemisia.argentifolia.nc?raw=true'

!wget  https://github.com/KateDuffy/climate-ecology-tutorial/blob/main/tas_day_EC-Eart
os.rename('tas_day_EC-Earth3_historical+ssp585_1850-2100_Bemisia.argentifolia.nc?raw=t

!wget  https://github.com/KateDuffy/climate-ecology-tutorial/blob/main/tas_day_INM-CM5
os.rename('tas_day_INM-CM5-0_historical+ssp585_1850-2100_Bemisia.argentifolia.nc?raw=t

# thermal performance data
!wget https://github.com/KateDuffy/climate-ecology-tutorial/blob/main/SD1.csv?raw=true
os.rename('SD1.csv?raw=true','SD1.csv')
```

```
Connecting to github.com (github.com)|140.82.114.4|:443... connected.
```

```
HTTP request sent, awaiting response... 302 Found
Location: https://github.com/KateDuffy/climate-ecology-tutorial/raw/main/tas_day_
--2022-04-17 14:55:20--  https://github.com/KateDuffy/climate-ecology-tutorial/ra
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/KateDuffy/climate-ecology-tutorial/ma
--2022-04-17 14:55:20--  https://raw.githubusercontent.com/KateDuffy/climate-eco
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.1
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.
HTTP request sent, awaiting response... 200 OK
Length: 1476241 (1.4M) [application/octet-stream]
Saving to: ‘tas_day_EC-Earth3_historical+ssp585_1850-2100_Bemisia.argentifolia.n


tas_day_EC-Earth3_h 100%[===================>]   1.41M  --.-KB/s    in 0.06s

2022-04-17 14:55:21 (25.4 MB/s) - ‘tas_day_EC-Earth3_historical+ssp585_1850-2100_

--2022-04-17 14:55:21--  https://github.com/KateDuffy/climate-ecology-tutorial/b
Resolving github.com (github.com)... 140.82.114.3
Connecting to github.com (github.com)|140.82.114.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github.com/KateDuffy/climate-ecology-tutorial/raw/main/tas_day_
--2022-04-17 14:55:21--  https://github.com/KateDuffy/climate-ecology-tutorial/ra
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/KateDuffy/climate-ecology-tutorial/ma
--2022-04-17 14:55:21--  https://raw.githubusercontent.com/KateDuffy/climate-eco
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.1
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.
HTTP request sent, awaiting response... 200 OK
Length: 1475265 (1.4M) [application/octet-stream]
Saving to: ‘tas_day_INM-CM5-0_historical+ssp585_1850-2100_Bemisia.argentifolia.n

tas_day_INM-CM5-0_h 100%[===================>]   1.41M  --.-KB/s    in 0.06s

2022-04-17 14:55:21 (25.1 MB/s) - ‘tas_day_INM-CM5-0_historical+ssp585_1850-2100_

--2022-04-17 14:55:21--  https://github.com/KateDuffy/climate-ecology-tutorial/b
Resolving github.com (github.com)... 140.82.112.4
Connecting to github.com (github.com)|140.82.112.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github.com/KateDuffy/climate-ecology-tutorial/raw/main/SD1.csv
--2022-04-17 14:55:21--  https://github.com/KateDuffy/climate-ecology-tutorial/ra
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/KateDuffy/climate-ecology-tutorial/ma
--2022-04-17 14:55:21--  https://raw.githubusercontent.com/KateDuffy/climate-eco
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.1
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.
HTTP request sent, awaiting response... 200 OK
Length: 5428 (5.3K) [text/plain]
Saving to: ‘SD1.csv?raw=true’

SD1.csv?raw=true    100%[===================>]   5.30K  --.-KB/s    in 0s

2022-04-17 14:55:22 (49.1 MB/s) - ‘SD1.csv?raw=true’ saved [5428/5428]
```

```
ls
```

```
sample_data/
SD1.csv
tas_day_BCC-CSM2-MR_historical+ssp585_1850-2100_Bemisia.argentifolia.nc
tas_day_CESM2_historical+ssp585_1850-2100_Bemisia.argentifolia.nc
tas_day_EC-Earth3_historical+ssp585_1850-2100_Bemisia.argentifolia.nc
tas_day_INM-CM5-0_historical+ssp585_1850-2100_Bemisia.argentifolia.nc
```

Reading data using xarray

```
model_1 = xr.open_dataset("tas_day_BCC-CSM2-MR_historical+ssp585_1850-2100_Bemisia.arg
model_2 = xr.open_dataset("tas_day_CESM2_historical+ssp585_1850-2100_Bemisia.argentifo
model_3 = xr.open_dataset("tas_day_EC-Earth3_historical+ssp585_1850-2100_Bemisia.arger
model_4 = xr.open_dataset("tas_day_INM-CM5-0_historical+ssp585_1850-2100_Bemisia.arger

# convert time dimension to plottable format
for model in [model_1, model_2, model_3, model_4]:
  try:
    datetimeindex = model.indexes['time'].to_datetimeindex()
    model["time"] = datetimeindex
  except:
    pass

model_1 = model_1.sel(time=slice("1850", "2099"))
model_2 = model_2.sel(time=slice("1850", "2099"))
model_3 = model_3.sel(time=slice("1850", "2099"))
model_4 = model_4.sel(time=slice("1850", "2099"))
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:9: RuntimeWarning: (
  if __name__ == '__main__':
```

```
print(model_1)
```

```
<xarray.Dataset>
Dimensions:  (time: 91250)
Coordinates:
    lat      float64 ...
    lon      float64 ...
  * time     (time) datetime64[ns] 1850-01-01T12:00:00 ... 2099-12-31T12:00:00
Data variables:
    tas      (time) float64 ...
```

```
model_1.lat, model_1.lon
```

```
(<xarray.DataArray 'lat' ()>
```

```
    array(25.)
    Coordinates:
        lat      float64 25.0
        lon      float64 ..., <xarray.DataArray 'lon' ()>
    array(280.)
    Coordinates:
        lat      float64 25.0
        lon      float64 280.0)
```

Plotting temperature data

```
fig, ax = plt.subplots(figsize=(20, 5))
model_1.tas.plot()
plt.title("Daily temperature", size=15)
plt.show()
```

Computing and plotting temperature statisitics

```
# mean temperature
# 1 (a)

for model in [model_1, model_2, model_3, model_4]:
  model.resample(time='10Y').mean().tas.plot(marker="o")
plt.title("Decadal mean of temperature", size=15)
plt.show()
```

```
# 1(b)
model_1t = model_1.sel(time=slice("1950", "2000"))
model_2t = model_2.sel(time=slice("1950", "2000"))
model_3t = model_3.sel(time=slice("1950", "2000"))
model_4t = model_4.sel(time=slice("1950", "2000"))

for model in [model_1t, model_2t, model_3t, model_4t]:
  model.resample(time='10Y').std().tas.plot(marker="o")
plt.title("Decadal std.dev of temperature", size=15)
plt.show()
```

```
new1 = model_1.sel(time=slice("2050", "2099"))
new2 = model_2.sel(time=slice("2050", "2099"))
new3 = model_3.sel(time=slice("2050", "2099"))
new4 = model_4.sel(time=slice("2050", "2099"))

for i in [new1, new2, new3, new4]:
  i.resample(time='10Y').std().tas.plot(marker="o")
```

```
plt.title("Decadal std.dev of temperature", size=15)
plt.show()
```

```
# variance of temperature

for model in [model_1, model_2, model_3, model_4]:
  model.resample(time='10Y').var().tas.plot(marker="o")
plt.title("Decadal mean of temperature", size=15)
plt.show()
```

```
import rpy2.robjects.numpy2ri
rpy2.robjects.numpy2ri.activate()
```

```
for model in [model_1, model_2, model_3, model_4]:
```

```
f_model = [rstats.spectrum(x[1].tas.values, plot=False)[0] for x in model.resample(t
s_model = [rstats.spectrum(x[1].tas.values, plot=False)[1] for x in model.resample(t
B = [np.polyfit(np.log10(f_model[i]), np.log10(s_model[i]), deg=1)[0] for i in range
plt.plot([x[0] for x in model.resample(time='10Y')][1:], B[1:])
```

Detecting climate trends

two-sided t test

```
historical = model_2.sel(time=slice("1950", "2000"))
future = model_2.sel(time=slice("2050", "2100"))
print(historical)
```

```
<xarray.Dataset>
Dimensions:  (time: 18615)
Coordinates:
    lat      float64 ...
    lon      float64 ...
  * time     (time) datetime64[ns] 1950-01-01 1950-01-02 ... 2000-12-31
Data variables:
    tas      (time) float64 ...
```

```
x = historical.resample(time='1Y').std().tas
y = future.resample(time='1Y').std().tas
```

```
plt.hist(x, alpha=0.5, label="historical")
plt.hist(y, alpha=0.5, label="future")
plt.legend()
plt.show()
```

```
# test for normality
for data in [x,y]:
  stat, p = shapiro(data)

  alpha = 0.05
  if p > alpha:
    print('Sample looks Gaussian (fail to reject H0)')
  else:
    print('Sample does not look Gaussian (reject H0)')
```

```
Sample looks Gaussian (fail to reject H0)
Sample looks Gaussian (fail to reject H0)
```

```
# Two-sided t-test. Null hypothesis: the 2 independent samples have identical average
t, p = stats.ttest_ind(x, y)
print("t(df=%s) = %s, p = %s" %((len(x) + len(y) - 2), np.round(t,2), np.round(p, 4)))

alpha = 0.05
if p > alpha:
 print('Samples have identical means (fail to reject H0)')
else:
 print('Samples do not have identical means (reject H0)')
```

```
t(df=99) = -2.61, p = 0.0106
Samples do not have identical means (reject H0)
```

Generalized least squares regression

```
import statsmodels.api as sm
gls_model = sm.GLS(endog=[x[0].astype('datetime64[Y]').astype(int)+1970 for x in model
gls_results = gls_model.fit()
print(gls_results.summary())
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: FutureWa
  import pandas.util.testing as tm
                                 GLS Regression Results
=================================================================================
Dep. Variable:                      y   R-squared (uncentered):
Model:                            GLS   Adj. R-squared (uncentered):
Method:                 Least Squares   F-statistic:                          2.
Date:                Sun, 17 Apr 2022   Prob (F-statistic):                    5
Time:                        14:55:29   Log-Likelihood:                        .
No. Observations:                  25   AIC:
Df Residuals:                      24   BIC:
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1         -1593.2485     10.843   -146.938      0.000   -1615.627   -1570.870
==============================================================================
Omnibus:                        0.456   Durbin-Watson:                   1.718
Prob(Omnibus):                  0.796   Jarque-Bera (JB):                0.245
Skew:                          -0.233   Prob(JB):                        0.885
Kurtosis:                       2.864   Cond. No.                         1.00
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
```

Exploring thermal performance data

```
TPC_data = pd.read_csv("SD1.csv")
print(len(TPC_data))
TPC_data.head()
```

$$P(T) = \begin{cases} \exp\left\{-\left(\dfrac{T - T_{\text{opt}}}{2\sigma_{\text{p}}}\right)^2\right\} & \text{for } T \le T_{\text{opt}} \\ 1 - \left(\dfrac{T - T_{\text{opt}}}{T_{\text{opt}} - CT_{\max}}\right)^2 & \text{for } T > T_{\text{opt}} \end{cases}$$

```
def TPC_growth_rate_func(t, T_opt, sigma, T_max):
    if t <= T_opt:
        r = np.exp( - ( (t - T_opt) / (2 * sigma) ) ** 2)
    elif t > T_opt:
        r = 1 - ( (t - T_opt)/(T_opt - T_max) ) ** 2
    return r

TPC_growth_rate = np.vectorize(TPC_growth_rate_func)
```

```
species = "Bemisia.argentifolia"
row = TPC_data[TPC_data["Species"] == species]
row
```

```
# plot a thermal performance curve

x = np.arange(-20,40,1) #temperatures from -20 to 40 C
y = TPC_growth_rate(x, row.Topt, row.SIGMA, row.CTmax)


fig, ax = plt.subplots(2,1, figsize=(10,10))
ax[0].plot(x, y)
ax[0].plot((-20, 50), (0, 0), "--", c="black", lw=1)
ax[0].set_ylim(-0.5, 1.15)
ax[0].set_xlim(-20, 45)
ax[0].set_ylabel("Growth rate", size=15)
ax[0].set_title("Thermal performance curve: %s" %(species), size=15)


a = [row.Topt.item() - 4*row.SIGMA.item(), row.Topt, row.CTmax]
b = [TPC_growth_rate(v, row.Topt, row.SIGMA, row.CTmax) for v in a]
n = ["Tmin", "Topt", "Tmax"]
ax[0].scatter(a, b, c="black")
for i, txt in enumerate(n):
    ax[0].annotate(txt, (a[i], b[i]), size=15)



ax[1].plot(x, np.gradient(np.gradient(y)))
b=[25]
ax[1].plot(b,0, marker='o')
ax[1].plot((-20, 50), (0, 0), "--", c="black", lw=1)
ax[1].set_ylim(-0.004, 0.004)
ax[1].set_xlim(-20, 45)
```

```
ax[1].set_ylabel("f \"(Growth rate)", size=15)
ax[1].set_xlabel("Temperature (C)", size=15)

plt.tight_layout()
plt.show()
```

the below graph is the second derivative of the growth rate. The effect of variation is beneficial when the 2nd derivative is positive.

```
print(TPC_growth_rate(t=10, T_opt=row.Topt, sigma=row.SIGMA, T_max=row.CTmax))
```

```
[0.00011323]
```

```
# calculating the temperature-dependent growth rate using synthetic data

temperature = np.random.normal(loc=20, scale=2, size=200)
r = TPC_growth_rate(temperature, row.Topt, row.SIGMA, row.CTmax)

fig, ax = plt.subplots(1,2, figsize=(16,5))
ax[0].plot(temperature)
ax[0].set_ylabel("Temperature (degrees C)", size=15)
ax[0].set_xlabel("Time step", size=15)

ax[1].plot(r)
ax[1].set_ylim(0,1.1)
ax[1].set_title("Mean growth rate = %s" % np.round(np.mean(r), 3), size=15)
ax[1].set_ylabel("Growth rate", size=15)
ax[1].set_xlabel("Time step", size=15)
plt.show()
```

```
# Demonstrating nonlinear averaging with synthetic data

fig, ax = plt.subplots(1,1, figsize=(8,5))

for sdev in np.arange(0, 5, 0.5):
  temperature = np.random.normal(loc=25, scale=sdev, size=5000)
  r = TPC_growth_rate(temperature, row.Topt, row.SIGMA, row.CTmax)
  plt.scatter(sdev, r.mean(), c="blue")

plt.ylabel("Average growth rate", size=15)
```

```
plt.xlabel("Standard deviation of temperature", size=15)
plt.show()
```

As the std. dev increases, the growth rate, performance of the species decreases.

Quantifying ecological fitness based on climate projections

```
fig, ax = plt.subplots(1,1, figsize=(10,5))

for model in [model_1, model_2, model_3, model_4]:
  r =  TPC_growth_rate(model.tas-273.15, row.Topt, row.SIGMA, row.CTmax)
  model["r"] = (("time"), r)

  model.resample(time='1Y').std().r.plot()
  plt.ylim(-0.25,1.1)

plt.title(species, size=15)
plt.ylabel("Growth rate", size=15)
plt.show()
```

The annual std. changes between 1950 and 2000 remians more or less the same.

The annual std. changes between 2050 and 2100. It it seen to slightly decrease with time. The 3 models indicated by the orange, green and blue lines are seen to decrease.

Modeling population dynamics using temperature-dependent growth rate

```
def r_alpha(t, y, pars):
    N = y[0]
    if N < 1e-9:
        dN = 0.
    else:
        loc = np.int(np.floor(t))
        rs = pars['r']
        r_current = rs[loc] + (t-loc) * (rs[loc + 1] - rs[loc])
        dN = N * (r_current - pars['alpha'] * (N ** pars['theta']))
    return([dN])


def run_r_alpha(r, theta = 1., alpha = 1.):
    t_beg = 0
    t_end = len(r) - 2
    t_span = [t_beg, t_end]
    times = np.linspace(t_beg, t_end, len(r))
    pars = dict(r = r, alpha = alpha, theta = theta)
    N0 = np.max([0.01, 1/alpha * r[0]])
    init_conds = [N0]
    sol = inte.solve_ivp(lambda t,y: r_alpha(t, y, pars),
                         t_span, init_conds, method = 'RK45', t_eval = times)
    return sol.y[0, :]
```

```
fig, ax = plt.subplots(1,1, figsize=(10,5))

for i, model in enumerate([model_1, model_2, model_3, model_4]):
  r =  TPC_growth_rate(model.tas-273.15, row.Topt, row.SIGMA, row.CTmax)
```

```
    N = run_r_alpha(r)
    model["N"] = (("time"), N)

    model.resample(time='1Y').mean().N.plot()
    plt.ylim(-0.25,1.1)

plt.title(species, size=15)
plt.ylabel("Population density", size=15)
plt.show()
```

2(d)Finally, calculate the mean air temperature from 2000-2100 for model_1 (don’t forget to convert from Kelvin to degrees Celcius). Describe the position of the mean temperature with respect to the thermal optimum, minimum, and maximum of Bemisia argentifolia. Bonus: Will temperature variability increase or decrease growth rate at this temperature?

```
model_1
```

```
#2d
model1_mean_temp = model_1.sel(time=slice("2000", "2100"))
ans = model1_mean_temp.tas.mean()
print("Mean temperature in K",ans )

#convert ans from K to Celcius
res = ans - 273.15
print("Mean temperature in Celcius for Bemisia argentifolia is:",res )
```

```
Mean temperature in K <xarray.DataArray 'tas' ()>
array(300.11113302)
Coordinates:
    lat      float64 25.0
    lon      float64 280.0
Mean temperature in Celcius for Bemisia argentifolia is: <xarray.DataArray 'tas'
array(26.96113302)
Coordinates:
    lat      float64 25.0
    lon      float64 280.0
```

```
# plot a thermal performance curve

x = np.arange(-20,40,1) #temperatures from -20 to 40 C
y = TPC_growth_rate(x, row.Topt, row.SIGMA, row.CTmax)


plt.plot(x, y)
plt.plot((-20, 50), (0, 0), "--", c="black", lw=1)
plt.ylim(-0.5, 1.15)
plt.xlim(-20, 45)
#ax[0].set_ylabel("Growth rate", size=15)
#ax[0].set_title("Thermal performance curve: %s" %(species), size=15)
a= [26]
b=[0.75]

plt.plot(a,b,marker = "o", markersize = 10, markerfacecolor='red')
```

```
a = [row.Topt.item() - 4*row.SIGMA.item(), row.Topt, row.CTmax]
b = [TPC_growth_rate(v, row.Topt, row.SIGMA, row.CTmax) for v in a]
n = ["Tmin", "Topt", "Tmax"]
plt.scatter(a, b, c="black")
for i, txt in enumerate(n):
    plt.annotate(txt, (a[i], b[i]), size=15)


plt.tight_layout()
plt.show()
```

Bonus: The temperature variability will not be beneficial to the growth rate at this temperature (the second derivative graph shows that 26C falls on the negative spectrum)

✓ 0s completed at 11:48 AM