ATSC 201 Example

September 16, 2025

0.1 Guidelines for non-Excel languages

- Your submission must be legible to **all** markers
- As with Excel, you must write out your thinking in **plain English**. You should also heavily comment your code
- Please stick to basic mathematical expressions and not advanced library functions. If you do use a library function, you must clearly state what it does and why you chose to use it.
- Your answers and any important intermediates (e.g. variables whose units have been converted) must be clearly stated and easy to find. This may mean printing out variable values more often than you normally would.
- Your submission must be in PDF format and should be clearly legible (no cut off images, etc.)
- If you fail to make your code clearly legible, you may miss out on points

1 ATSC 201 HW 2

1.0.1 Name: Ben Hartvigsen

1.0.2 SN: 12345678

1.0.3 Date: September 16, 2025

2 Chapter 1

1) Plot the change in density of dry air with height in an isothermal (constant temperature) atmosphere. Use pressure (P) coordinates as a surrogate for height, with the highest pressure at the bottom of the vertical (y) axis. Plot for pressures of 100 - 0 kPa, at 10 kPa increments.

a) use T = 25 degC, and plot on a linear-linear graph

b) copy the graph from (a) and change the vertical (P) axis to be logarithmic. (i.e., semi-log graph)

c) copy the graph from (b) and use it to plot multiple semi-log graph curves. In addition to T = 25 degC already plotted, add the curves for T = 0 degC and T = -25 degC.

2.1 Question 1a

a) use T = 25 degC, and plot on a linear-linear graph

2.1.1 Given

P_range = 100 to 0 kPa

Dry air

T = 25 degC

2.1.2 Find

rho vs. P = ? kg/m3

2.1.3 Assume

Ideal gas law

2.1.4 Equation

P = rho * Rd * T

Solving for rho,

rho = P/(Rd * T)

[ ]:
```
# Library imports
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

[ ]:
```
### Given Variables
T_C = 25 #degC
P = np.arange(100, -1, -10) # kPa (this creates an array of the pressure range)

print('P =', P, 'kPa')

### Constant Variables
Rd = 0.287053 #kPa * K-1 * m3 * kg-1
```

```
P = [100  90  80  70  60  50  40  30  20  10   0] kPa
```

[ ]:
```
### Convert temperature
T_K = T_C + 273.15 # K

print('T_K =', T_K, 'K')
```

```
T_K = 298.15 K
```

[ ]:
```
### Calculate rho values
rho = P / (Rd*T_K) # kg/m3
```

[ ]:
```
### Print table of P and rho values
# dataframe is being used to create a table
```

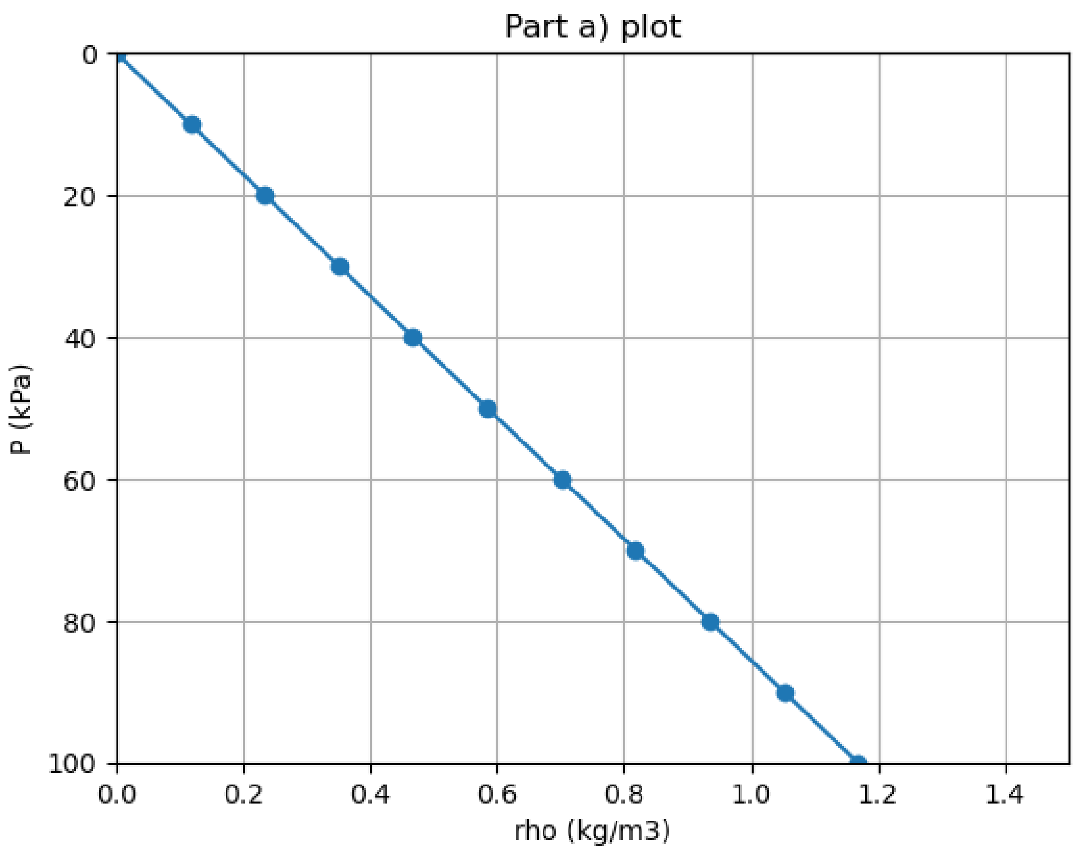
```python
df = pd.DataFrame({
    'P (kPa)': P,
    'rho (kg/m3)': rho
})

df
```

[ ]:

	P (kPa)	rho (kg/m3)
0	100	1.168431
1	90	1.051588
2	80	0.934745
3	70	0.817902
4	60	0.701059
5	50	0.584216
6	40	0.467372
7	30	0.350529
8	20	0.233686
9	10	0.116843
10	0	0.000000

[ ]:

```python
### Plot P against rho
plt.plot(rho, P, marker='o', linestyle='-')

# Create axis labels
plt.ylabel('P (kPa)')
plt.xlabel('rho (kg/m3)')
plt.title('Part a) plot')

# Set axis limits
plt.xlim([0,1.5])
plt.ylim([0,100])

# Invert y-axis
plt.gca().invert_yaxis()

# Include grid for legibility
plt.grid()

plt.show()
```

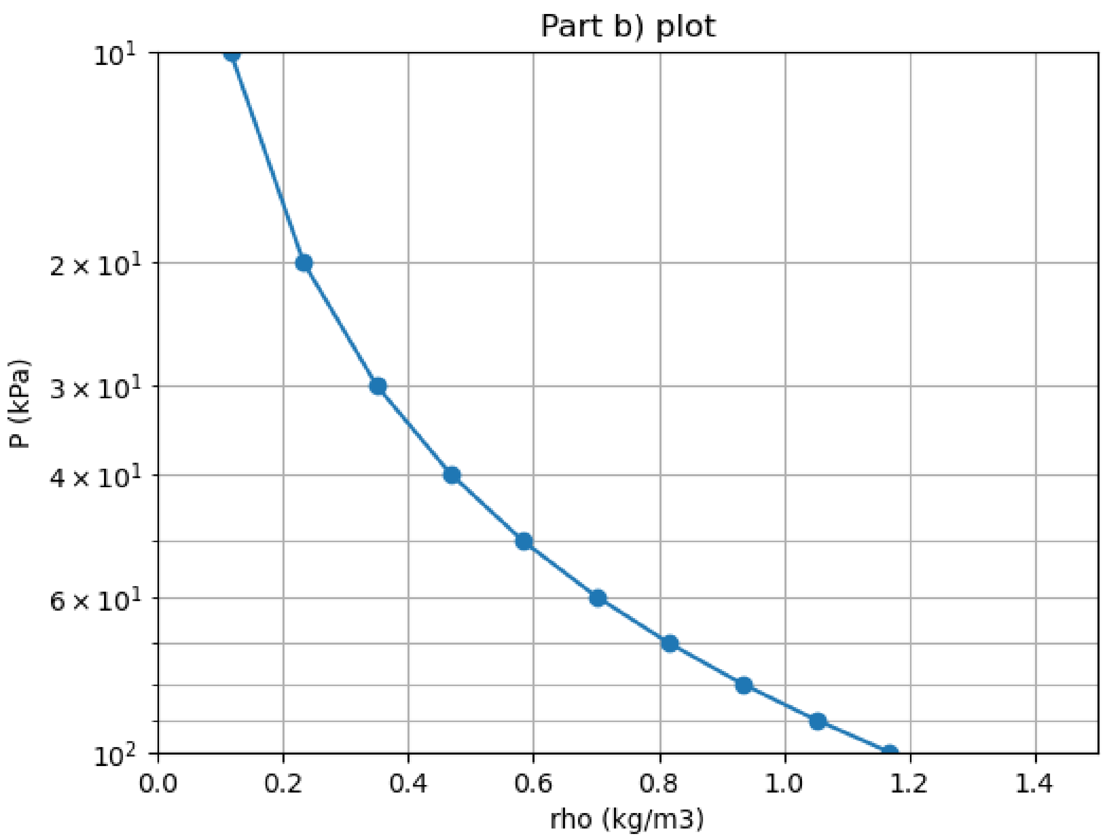


2.2 Question 1b

b) copy the graph from (a) and change the vertical (P) axis to be logarithmic. (i.e., semi-log graph)

[ ]:
```
### Plot P against rho
plt.plot(rho, P, marker='o', linestyle='-')

# Set y scale to be logarithmic
plt.yscale('log') # <- this is the change

# Create axis labels
plt.ylabel('P (kPa)')
plt.xlabel('rho (kg/m3)')
plt.title('Part b) plot')

# Set axis limits
plt.xlim([0,1.5])
plt.ylim([10,100]) # <- lower limit changed from 0 to 10
```

```
# Invert y-axis
plt.gca().invert_yaxis()

# Include grid for legibility
plt.grid(which='both')

plt.show()
```



2.3 Question 1c

c) copy the graph from (b) and use it to plot multiple semi-log graph curves. In addition to T = 25 degC already plotted, add the curves for T = 0 degC and T = -25 degC.

[ ]:
```
### New givens
TC_0 = 0
TC_m25 = -25
```

[ ]:
```
### Convert new temperatures
TK_0 = TC_0 + 273.15
```

```python
TK_m25 = TC_m25 + 273.15

print('0 degC =', TK_0, 'K')
print('-25 degC =', np.round(TK_m25,3), 'K')
```

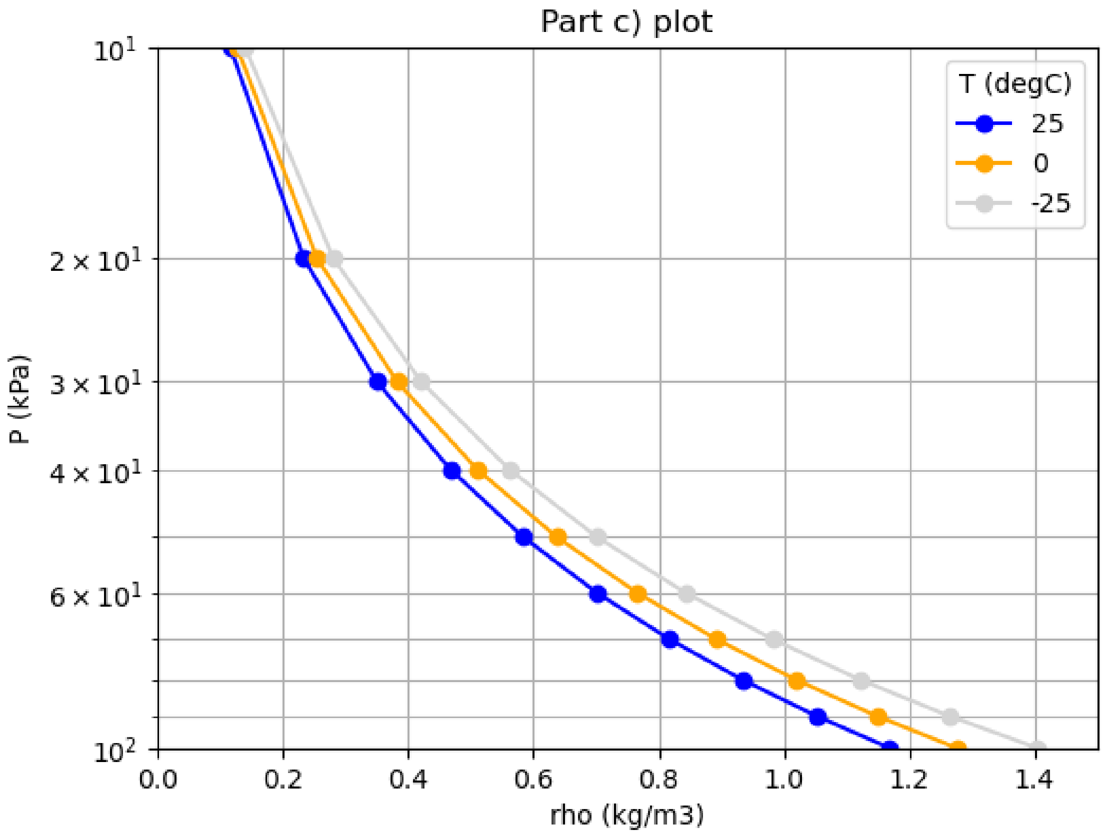
```
0 degC = 273.15 K
-25 degC = 248.15 K
```

[ ]:
```python
### Calculate new densities
rho_0 = P/(Rd*TK_0)
rho_m25 = P/(Rd*TK_m25)
```

[ ]:
```python
### Create new table for P and rho
# dataframe is being used to create a table
df_new = pd.DataFrame({
    'P (kPa)': P,
    'rho (kg/m3), T=25degC': rho,
    'rho (kg/m3), T=0degC': rho_0,
    'rho (kg/m3), T=-25degC': rho_m25
})

df_new
```

[ ]:
```
    P (kPa)  rho (kg/m3), T=25degC  rho (kg/m3), T=0degC  \
0       100               1.168431              1.275371
1        90               1.051588              1.147834
2        80               0.934745              1.020297
3        70               0.817902              0.892760
4        60               0.701059              0.765223
5        50               0.584216              0.637686
6        40               0.467372              0.510149
7        30               0.350529              0.382611
8        20               0.233686              0.255074
9        10               0.116843              0.127537
10        0               0.000000              0.000000

    rho (kg/m3), T=-25degC
0                 1.403859
1                 1.263474
2                 1.123088
3                 0.982702
4                 0.842316
5                 0.701930
6                 0.561544
7                 0.421158
8                 0.280772
9                 0.140386
10                0.000000
```

[ ]:
```python
### Recreate plot with all three curves
plt.plot(rho, P, marker='o', linestyle='-', label='25', color='blue') # Plot␣
↪25degC curve
plt.plot(rho_0, P, marker='o', linestyle='-', label='0', color='orange') # Plot␣
↪0degC curve
plt.plot(rho_m25, P, marker='o', linestyle='-', label='-25', color='lightgrey')␣
↪# Plot 0degC curve

# Set y scale to be logarithmic
plt.yscale('log')

# Create axis labels
plt.ylabel('P (kPa)')
plt.xlabel('rho (kg/m3)')
plt.title('Part c) plot')

# Set axis limits
plt.xlim([0,1.5])
plt.ylim([10,100])

# Invert y-axis
plt.gca().invert_yaxis()

# Include grid for legibility
plt.grid(which='both')

# Create legend
plt.legend(title='T (degC)')

plt.show()
```



2.3.1 Check

Units OK. Curves approximately agree with Table 1-5

2.3.2 Discussion

The air is thinner at higher altitudes (where the pressure is lower). So each breath of air would bring in fewer oxygen molecules at higher altitudes, causing me to become hypoxic.