# Nick McCullough

Project 1 Report
AerE 161
Due Date: 3/4/2022 6pm CST

# Contents


| | |
|---|---|
| Table of Contents | 2 |
| Problem Statements | 3 |
| Theory | 4 |
| Solution | 6 |
| Discussion | 7 |
| Challenges | 8 |
| Appendix | 9 |

# Problem Statements

For problem 1, we will do a weight and balance check for a 1965 Piper Cherokee PA-28-180 aircraft. This aircraft has four seats with a Lycoming O-360-A3A engine.

Given information of this aircraft:

Empty Weight is 1,471 lbs.
Center of Gravity (empty) is 85.9 in
Front Seats Moment Arm is 85.5 in
Fuel Tanks Moment Arm is 95 in
Rear Seats Moment Arm is 118.1 in
Maximum ramp weight is 2,400 lbs.
Maximum fuel capacity is 50 US Gallons

We will need to write a script and function that obtains missing information from the user, Define variables used with the information given, Calculate the fuel weight, Calculate the ramp weight, Calculate the moment arm for each area of the aircraft, Calculate the Center of Gravity (CG) of the aircraft, Determine if the aircraft is within it's weight and balance and then print out the ramp weight, loaded moment and if the aircraft is within weight and balance.

For problem 2, we will calculate the density altitude using real-time data. We will need to create a function the reads the API from Openweathermap.org. We can access real time local data from this API such as temperature, pressure, and humidity.

We will need to convert from Kelvin Temp to Celsius, tempC. Then, find the Dewpoint temperature (TD). Once we have the Dewpoint Temperature, we can calculate the vapor pressure. Vapor pressure allows us to calculate the virtual temp and then convert that into Rankine temperature. Finally, we can calculate the density altitude using a conversion from millibar pressure to pressure in Mercury. We use the pressure in Mercury and Rankine virtual temperature plus our local field elevation in Ames, to find our density altitude.

The pilot will be able to make the decision, based off this reading, if the density altitude is safe enough to fly the aircraft.

# Theory

Instructions Equation #

**Equations used in Project 1, Problem 1.**

*moment = weight(lbs.) * arm(in)* Equation 1.

This equation calculates the moment of a particular point in the aircraft

$$CG_{aircraft} = \frac{moment_{total}}{aircraft_{totalweight}} \quad \text{Equation 2.}$$

This equation calculates the new center of gravity using rampweight and total moment.

*rampweight = emptyweight + fuelweight + pilot + copilot + pass1 + pass2*

This equation calculates the total aircraft weight with user input variables

*fuelweight=fuel*6*

This is our fuel weight; we multiply by six because each gallon is 6 lbs.

**Equations used in Project 1, Problem 2.**

*tempC = temp – 273.15*

This is the equation used to convert from Kelvin to Celsius. From the API, the temperature is in Kelvin.

$$e = 6.11 \cdot 10^{\left(\frac{7.5 \cdot T_d}{237.7 + T_d}\right)} \quad \text{Equation 3.}$$

This is the equation for vapor pressure.

$$T_d = T_c - \left(\frac{100 - Humidity}{5}\right) \quad \text{Equation 4.}$$

This is the equation for dewpoint in the atmosphere. $T_d$ is the dewpoint temperature, while $T_c$ is the temperature in Celsius. $Humidity$ is real time humidity percentage in Ames.

$$T_v = \frac{T}{1 - \left(\frac{e}{P_{mb}}\right) \cdot (1 - 0.622)} \quad \text{Equation 5.}$$

This is the equation used to calculate virtual temperature. $T$ is the current temperature in Kelvin. $P_{mb}$ is the current pressure using millibar.

$$\left(\frac{9}{5}\cdot\left(T_v - 273.15\right) + 32\right) + 459.69 \qquad \text{Equation 6.}$$

This equation is how we convert from Kelvin temperature to Rankine temperature.

$$D_{alt} = FieldElevation + \left(145366\cdot\left(1-\left(\frac{17.326\cdot P_{inHG}}{T_{vr}}\right)^{0.235}\right)\right) \qquad \text{Equation 7.}$$

This equation is how we solve for Density Altitude. We use our local field elevation from the Ames airport and plug in the pressure in Mercury as well as our virtual temperature.

$$P_{inHG} = P_{mb}\cdot 0.02953 \qquad \text{Equation 8.}$$

This equation converts our Pressure in millibar to Pressure in Mercury.

# Solution

My approach to solving the problems for this project were to read through the instructions and type out the code as I understood it. For example, I coded the equations for Problem 2 as I was reading it. This allowed me to get a good start on what I needed to do once I had all the equations coded.

**Weight and Balance Values:**

Fuel = 40 gallons
Pilot = 120 lbs.
Copilot = 120 lbs.
Pass1 = 120 lbs.
Pass2 = 120 lbs.

## Output for Problem 1:

```
Hello, welcome to the Weight and Balance Calculator.
This aircraft: 1965 Piper Cherokee PA-28-180
How much fuel, in US Gallons, is on board the airplane?: 40
Please enter the weight of the pilot: 120
Please enter the weight of the co-pilot: 120
Please enter the weight of passenger 1, if no passenger 1 enter 0: 120
Please enter the weight of passenger 2, if no passenger 2 enter 0: 120
The aircraft is within the maximum weight.
The Total Moment of the aircraft is : 179022.90
Aircraft is balanced. OK to fly.



 Finished?
 Enter 1 to exit.
1
Thank you for using the Weight and Balance Calculator.
>>
```

## Output for Problem 2:

The Temperature in degrees Celsius is

tempC =

    9.2700

The current density altitude is : 43157.390925 feet

>>

# Discussion

**– Analysis and interpretation of results.**

All passengers weighed the same, 120 lbs. Using common sense, this seems to be a good balance for passenger weight. Used 40 gallons of fuel to counter the weight of the passengers. I was previously using 25 gallons, and the aircraft was unbalanced. The rampweight being 1991 lbs is quite a bit under the limit of MaxRampWeight of 2,400 lbs. The TempC in problem 2 was 9.27 degrees is around 48 degrees Fahrenheit, which seemed correct for today (Mar 4) – not too hot. The density altitude I calculated was 42,967 feet. I googled the average range for airplane flight in feet, which was 33k feet – 42k feet. Considering Ames airport is 955.6 feet above the sea level, I feel this is accurate and my calculations are correct.

**– Did this result in the aircraft being able to fly?**

Yes, the Aircraft is balanced and within the weight limit and conditions are good.

**– Why is weight and balance important in aviation?**
It is very important because it is the variable you can control on the airplane. You can control how much weight goes in the airplane, where the weight goes and you (the pilot) are the one that determines if the plane is balanced, making it stable enough to fly. If the plane is too heavy, it won't take off. This is important information to know ahead of flying the plane.

**– What was the density altitude that you calculated?**

42,967 feet

**– Why is density altitude important in aviation?**
Density altitude is another important thing as mentioned above with weight and balance. Weight and balance is within the pilot's control, however, density altitude is not. Density altitude changes with

the weather. The pilot can control his decision making before takeoff. That is the part in the pilot's control.

High density altitude, when the air is hot and humid, can cause significant issues during takeoff and landing because it affects the airplane's lift and engine performance. The air is denser towards the surface of the earth. The greater the altitude the lower the pressure and density, as the air molecules are more spread out. Pressure altitude corrects the altitude based on the current pressure. Density altitude takes the pressure altitude and corrects it for temperature. The higher the density altitude number, confirms that we have poorer flying conditions.

Therefore, it is important to calculate the density altitude, to ensure it is optimal for flight. The pilot should calculate this before flying, it is just as important as weight and balance.

# Challenges

I had some challenges with this project. This is the largest programming project I have ever done, and I am new to programming. I enjoyed it and it was fun figuring things out. I had some trouble with my if statements in the code in WBcalc and understanding exactly what to use in the function and in the script. Once I figured this out, it wasn't so bad. I had all the equations and inputs already in my script, so it was just figuring how to organize it properly between the two.

When I ran into these issues, I reached out to TA Sravya Sasetty via Telegram who reviewed my code and helped me with next steps. This helped me a lot. It turns out I had done *too much* work and needed to fix very minor things. My equations were all correct. I am glad I had all my equations ready when I asked for help, as it went very smoothly figuring out where I went wrong. This all comes back to my approach as mentioned earlier in the Solution, it helped that I coded all the equations as I read through the instructions, because I best understood them in real time.

# Appendix

```
% Nick McCullough, AerE 161, Project 1, Problem 1 function

function [fuel,pilot,copilot,pass1,pass2] = Problem1
% create function "Problem1" using variables below

fuel = input('How much fuel, in US Gallons, is on board the airplane?: '); % input statement
while (fuel<0) || fuel> 50 % create while statement for bounds
   disp('Error! Please Try Again. Fuel must be between 0 and 50 gallons.') % error statement
   fuel = input(); % requires input
end % end while statement

pilot = input('Please enter the weight of the pilot: '); % user input
copilot = input('Please enter the weight of the co-pilot: '); % user input
pass1 = input('Please enter the weight of passenger 1, if no passenger 1 enter 0: '); % user input
pass2 = input('Please enter the weight of passenger 2, if no passenger 2 enter 0: '); % user input

end % end function
```

---

```
% Nick McCullough, AerE 161, Project 1, Problem 1 Script
```

```
% created a function called "Problem1" for the weights of user inputs

clear,clc % clears all code
% Start by entering all data for the aircraft.

% % we will be storing all values as variables listed in Instructions Table 1
emptyweight = 1471; % this is the empty weight of the airplane, pounds, fixed
cg = 85.9; % this is the center of gravity of the airplane, inches, fixed
FSMA = 85.5; % this is the Front Seats Moment Arm, inches
FTMA = 95; % this is the Fuel Tanks Moment Arm, inches
RSMA = 118.1; % this is the Rear Seats Moment Arm, inches
MaxRampWeight = 2400; % this is the Maximum Ramp Weight, pounds (cannot exceed this)
MaxFuel = 50; % this is the Maximum Fuel Capacity, US Gallons, 100LL fuel

while(1) % this is for while loop and exit script at the end.

disp('Hello, welcome to the Weight and Balance Calculator.') % simple welcome greeting
disp('This aircraft: 1965 Piper Cherokee PA-28-180') % naming the aircraft

[fuel,pilot,copilot,pass1,pass2] = Problem1; % calling the function Problem1
rampweight = (emptyweight + fuel + pilot + copilot + pass1 + pass2);
% calculate rampweight using Problem 1 value, adding emptyweight value


  if rampweight > MaxRampWeight % create if statement for current rampweight and MaxRampWeight
variable
  disp('The plane is too heavy to fly. Please remove weight.') % too heavy :(
  elseif rampweight <= MaxRampWeight % else if statement, less than or equal to
  disp('The aircraft is within the maximum weight.') % within weight
  else % else
  disp('Error. Please try again.') % Error statement for else
  end % end if statement
```

```
momentfront = (pilot + copilot)*FSMA; % using our rampweight function to find front seat moment
momentrear = (pass1 + pass2)*RSMA; % using our rampweight function to find rear seat moment
momentfueltank = (fuel)*FTMA; % using our rampweight function to find fuel tank moment
momentempty = (emptyweight)*85.9; % using our rampweight function to find empty moment
totalmoment = (momentfront + momentrear + momentfueltank + momentempty); % sum of moments

fprintf('The Total Moment of the aircraft is : %.2f\n',totalmoment)
% fprintf statement for the Total Moment

x = totalmoment; % variable for total moment
cgcalc = x / rampweight; % created variable to identify the calculated cg
% calculated center of gravity is x / rampweight

% per instructions, cgcalc should be between 85.8 and 95.8

  if cgcalc < 85.8 || cgcalc > 95.8 % calculated cg bounds statement per Instructions
  disp('Center of Gravity is off, aircraft not balanced')
  else
  disp('Aircraft is balanced. OK to fly.')
  end % end if statement

Z = input('\n\n Finished? \n Enter 1 to exit. \n'); % exit script for While(1) above
  if Z == 1 % exit variable 1 per instructions
    disp('Thank you for using the Weight and Balance Calculator.') % thank you
   break % break to exit
  end % end Finished?
end % end while(1)
```

```
% Nick McCullough, AerE 161, Project 1, Problem 2 function

function tempC = Problem2(temp) %create function "Problem2" Kelvin to Celsius

tempC = temp - 273.15; % conversion equation for given, Kelvin, to Celsius.
end % end function
```

---

```
% Nick McCullough, AerE 161, Project 1, Problem 2 Script

clear,clc % clears all code
% created function called "Problem2" converting Kelvin to Celsius

key = '7cab1fcaf444883263bc48dd983e6018'; % API Key
options = weboptions('ContentType','json');
url=['https://api.openweathermap.org/data/2.5/weather?q=','Ames','&APPID=',key];
CurrentData = webread(url, options);

temp = CurrentData.main.temp; % real time temp
pressure = CurrentData.main.pressure; % real time pressure
humidity = CurrentData.main.humidity; % real time humidity%

disp('The Temperature in degress Celsius is') % simple disp for Celsius
tempC = Problem2(temp) % call the function

TD = tempC - ((100-humidity)/(5)); % Dewpoint Temp
% Equation 4.

vaporpressure = 6.11 * 10^((7.5*TD)/(237.7+TD)); % vapor pressure
% Equation 3. e is in millibar
```

```
virtualtemp = (temp)/(1-(vaporpressure/pressure)*(1-0.622));
% Equation 5. virtual temp equation. temp is in Kelvin

virtualtempR = ((9/5)*(virtualtemp - 273.15) + 32) + 459.69;
% Equation 6. Rankine virtual temp.

fieldelv = 955.6;
% Ames Airport Elevation in feet

pressureinHG = pressure * 0.02953;
% Equation 8. Pressure in Mercury conversion from millibar.

densityaltitude = fieldelv + (145366*(1-(17.326 * pressureinHG)/virtualtempR)^0.235);
% Equation 7. Density altitude calculation.

fprintf('The current density altitude is : %f feet \n',densityaltitude)
% print statement to display density altitude.
```