# BEE313 Problem 1.1

## Objective

The objective of this problem was to determine the groundwater contribution to a 10km2 catchment in the Cascades. This was done by completing a water mass balance. The average precipitation is known to be 1750 mm/year and the average stream flow is .3 m3/sec. The ET is also known to be 4 mm/day. It was assumed that there was no change in storage. However, for this basin both the input and output groundwater flows were not known. By combining the two groundwater terms into one, the overall contribution to the basin was be determined as well as its direction.

### Equations:

Deriving Equation for GW:

- P + GWin- ET - Q -GWout = deltaS
- GWout - GWin = P - ET - Q - deltaS
- GW = P - ET - Q - deltaS

Assumption:

- deltaS = 0

## Methods:

First, developed a function for mass balance of the groundwater flow, and made note of all known parameters. The Stream flow was then divided by the area in order to get into a measurment of depth. Next, the precipitation, stream flow, and evapotranspiration values were converted to units of mm/year. The after converting to the proper units that values were plugged back into the function to return the groundwater flow in mm/year.

In [ ]:
```
import numpy as np

# function to solve for GW
def GW (P,Q,ET,dS):
    GW = P-ET-Q-dS
    return GW

# known parameters
rain_mm_yr = 1750 #mm/year
annual_strm_flow = .3 #m3/sec
avg_ET = 4 #mm/day
deltaS = 0 #no change
area =10*1000*1000*1000*1000 #mm2
```

```
# solved for Q
annual_strm_flow_mm = .3*1000*1000*1000 #mm3/s
Qmm_sec = annual_strm_flow_mm/area # mm/s

# convert all units to mm per year
## converted the stream flow into mm/year
Q_mm_yr= Qmm_sec*60*60*24*365
print(Q_mm_yr)
## converted the ET to mm/year
ET_mm_yr = avg_ET*365

# Applied the new parameters into the function
GW_final = GW(rain_mm_yr, Q_mm_yr,ET_mm_yr,deltaS )*-1

# Printed Answer
print(f"Groundwater Contribution GW_final {GW_final:.3f} mm/year into the basin")
```

```
946.08
Groundwater Contribution GW_final 656.080 mm/year into the basin
```

## Results:

The Groundwater contribution for the 10km2 basin in the Cascades was determined to be 289.999 mm/year out of the the basin. The precipitation had the largest contribution of 1750 mm/year. This was followed by the average ET with 1460 mm/year. The stream flow represented as a depth had a contribution of 946.08 mm/year, which was the smalled contribution.

## Discussion:

The results show that the majority of the the groundwater is moving out of the basin. It would make sense for the majority of the water to enter this basin because of the fact that due to gravity and pressure dynamics within the soil, the water will be forced to move down the hill to the lower regions towards where the water will eventually discharge (Massachusetts Audubon Society). The initial assumption was made that the change in the storage would be equal to zero. This assumption would be valid because within the annual hydrologic cycle the change in the amount of water held within the soil and in aquifers would be small since it is a cylical processes and it will start and end at the same point (NCRS, 2009).

# BEE313 Problem 1.2

## Objective:

The objective of this problem is to determine the minimum amount of rainfall that occurs from the Pacific Ocean to Mary's Peak. The Pacific Ocean had an air temperature of 21C and a relative humidut of 65%. Mary's Peak on the other hand had a temperature of 8C and a relative humidity that was not super saturated. The average thickness of the air mass is 2.5 km.

## Equations:

- Dew Point:
  - T_dp = (ln(e)-6.415)/(.0999-.000421*ln(e))
- Clausius–Clapeyron
  - e = *611*e^((17.3*T)/(T+237.3))
    - e* = Saturation Vapor Pressure
    - T = Temperature (C)
- Relative Humidity
  - RH = e/e*
    - e = vapor pressure
    - RH = relative humidity
- Vapor Density
  - pv = e/(Rv*T)
    - pv = vapor density
    - Rv = 461 J/kg*K
    - e = vapor pressure
- Precipitation Equation
  - P = [pv(t1)-pv(t2)]*h/pw
    - pw = density of water
    - h = parcel thickness
    - P = Precipitation ## Methods: The Clausisus-Clapeyron equation was used to calculate e* for 21C and 8C. These values were then plugged into their respective RH equations. The RH for 21C was 65%, and was used to find the vapor pressure. The dew point equation was then used to determine at what temperature condensation began to occur. The vapor pressure at 8C was found using the RH of 100%. The vapor pressures were then used to find the vapor density at each temperature. With the vapor densities at both 21C and 8C the precipitation equation was used to determine the minimum amount of rainfall.

In [ ]:
```
# Assumptions:
import numpy as np


# constants
Rv = 461 # J/kg*K
rho_water = 1000 #1000 kg/m^3
h = 2500 #m



# pv for 21C (t1)
e_sat21 = 611*np.exp(17.3*(21)/((21)+237.3))

## found vapor pressure
```

```
e_21 = .65*e_sat21

### found pv21
pv21 =  e_21/(Rv*(21+273.15))


# dew point temp
T_dp = (np.log(e_21)-6.415)/(.0999-.000421*np.log(e_21))
print(f"Dew Point Temperature: {T_dp: .3f} C")


# pv for 8C (t2)
## e*
e_sat8 = 611*np.exp(17.3*(8)/((8)+237.3))

## found vapor pressure
### Where RH = 1
e_8 = 1*e_sat8

### found pv8
pv8 =  e_8/(Rv*(8+273.15))

# Precipitation
P = (pv21 - pv8)*(h/rho_water)
P_m = P * 1000
print(f"Precipitation {P_m: .3f} mm")
```

```
Dew Point Temperature:  10.082 C
Precipitation  9.166 mm
```

## Results:

It was determined that the minimum amount of rainfall from the Pacific Ocean to Mary's Peak was 9.166 mm.

## Dicussion:

It was assumed that at Mary's Peak the relative humidity was 100%. This was based on the condition that it was not at supersaturation. It was known tha AT RH =1 it doesn't necessarily have to be raining but rather that is when the air is fully saturated. The given situation has a case where the air is not super saturated so in order to find the minimum amount given that the dew point was at 10C it is assumed that the RH remained the same from this dew point to the peak. The realtive humidity is typically inversely related to the temperature so as the temperature goes down, or as the air rise, then the relative humidity increases. In this case a simplification was made. If it was super saturated at the peak, then this would have decreased the level of precipitation.

# BEE313 Problem 1.3

## Objective:

The objective of this problem was to determine the average annual precipitation that the watershed recieved. The known elevation and precipiation data given at the gauge have been provided. The Area elevations and the fraction of the overall area that those elvations make up has also been provided in order to develop a predicted precipitation for those elevations.

GaugePrecipitation = [715, 750, 895]

GaugeElevations = [460, 600, 1180]

Topographic analysis of this watershed gives the following distribution of its area with elevation:

AreaFraction = [0.0, 0.028, 0.159, 0.341, 0.271, 0.151, 0.042, 0.008]

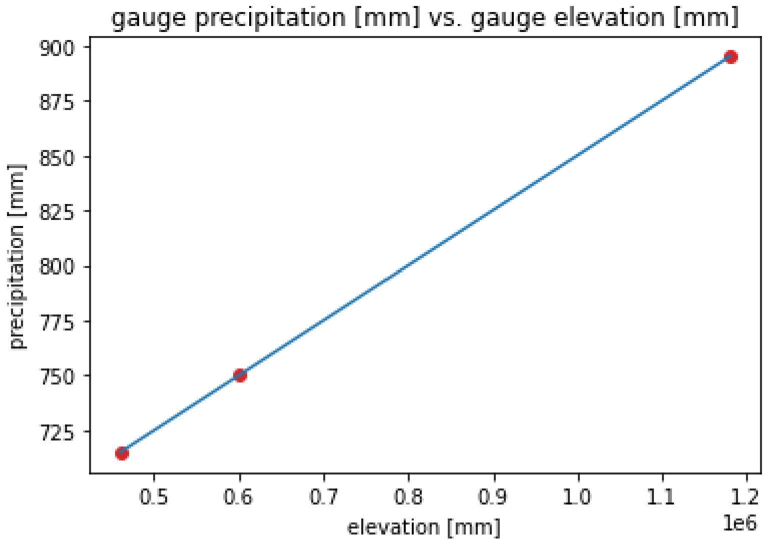
AreaElevations = [100, 300, 500, 700, 900, 1100, 1300, 1500]

## Methods:

A plot of the known precipitation and elevations from the stream gauge was developed. An equation representing these values on the plot was generated. Then using curvefit() a regression was run in order to determine the necessary coefficients for the model equation. With the model equation developed the area elevations were then inputted into the model to generate predicted precipitation values. Next, these precipitation values were weighted by multiplying the array by an array of the array fractions. Finally the weighted precipitation values were summed to get the average precipitation.

```
In [ ]: import matplotlib.pyplot as plt
        import pandas as pd
        import numpy as np
        from scipy.optimize import curve_fit
        # Step 1 inputted data into two separate dictionaries

        gauge_dict = {
            "P_gauge":[715,750,895],
            "E_gauge":[460,600,1180]
        }
        area_dict = {
            "area_fraction":[0.0,.028,.159,.341,.271,.151,.042,.008],
            "E_area":[100,300,500,700,900,1100,1300,1500]
        }
        gauge_df = pd.DataFrame(gauge_dict)
        area_df = pd.DataFrame(area_dict)
        # Step 2: performed a Regression on the Limited Data Available for Precipitation and G
        ## Set the figure

        ## Set the x and y values (observed gauge elevation and observed precipitation and )
        ### converted to numpy arrays
        obs_elv = (gauge_df['E_gauge'].values)*1000
        obs_precip =gauge_df['P_gauge'].values

        # Initial Plot of Points
        fig1 = plt.figure()
        ax3_1 = fig1.add_subplot(111)
        ax3_1.plot(obs_elv,obs_precip,color= "tab:blue", label = "observed")
```

```
ax3_1.scatter(obs_elv,obs_precip,color= "tab:red", label = "observed")
ax3_1.set_xlabel("elevation [mm]")
ax3_1.set_ylabel("precipitation [mm]")
ax3_1.set_title("gauge precipitation [mm] vs. gauge elevation [mm]")
plt.show()

# Step 3:Created a Function for Modeling the Precipitation
### This was constructed based on the determiningation that the precipirtion follows a
def modelP(elevation, P_const, P_init):
     precip = P_const*elevation + P_init
     return precip



# Step 4: Used Curve Fit to determine the necessary coefficients needed for this model
## Data To Input
elv_area = (area_df['E_area'].values)*1000
### Curvefit(model equation, x observed values, y observed valyes, coefficients in a t
## generated coefficients for model
popt,pcov = curve_fit(modelP, obs_elv, obs_precip, p0=(5,1))

## then using the new model parameters the estimated non-weighted precipitation was fo
est_precip = modelP(elv_area, popt[0], popt[1])

# Step 5: Incorporate Weights to the Precipitation Values
## saved these estimated precipitation values into a new column in the dataframe
area_df['est_p_wght'] = est_precip
est_p_no_wght = area_df['est_p_wght']

## Apply the weight to the precipitation values and change the name of the column in t

# #### Multiply Area Fraction (Weight) By Precipitation Values
area_df['est_p_wght'] =  est_p_no_wght * area_df["area_fraction"]
est_p_wght = area_df['est_p_wght']
# Step 6: Now will sum the values
Ann_Avg_Precip = np.sum(est_p_wght)

print(f"Average Annual Precipitation {Ann_Avg_Precip:.3f} mm/year")
```



```
Average Annual Precipitation 800.800 mm/year
```

## Results:

The average annual precipitation was found to be 800.8 mm/year.

## Discussion:

Rather than taking the predicted precipitation values and then averaging them over each of the recorded elevations a weight was taken into account. This was to ensure that the distribution of precipitation values across the elevations truly represented the region overall. Since the area elevations are only a sample of the elevations within this region the fraction helps to account to the wider population instead of this single sample. Then by summing these values the average precipitation within this basin can be found. If this were to be implemented in the future a larger dataset of observed values would have been useful. Then this dataset could have been split into a test and training datasets. However, for the purpose of the current model this is not necessary for future implementation this may be something that might need to be performed.

# BEE313 Problem 1.4

## Objective:

Using the annual rainfall data collected from 1901 to 1999 and a weibull plotting position the 10-year annual precipiation was determined for Corvallis in mm.

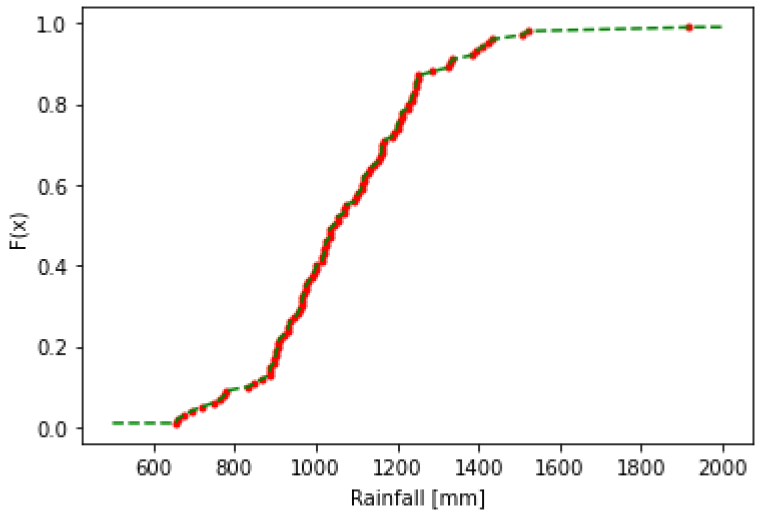
AnnualRainfall =
[1021.06,1052.48,1330.55,889.06,1407.20,886.63,1163.28,1168.88,897.59,1249.29,
993.65,963.82,1240.87,958.52,1144.11,1017.51,1252.29,1074.10,897.10,1284.55,1097.71,1105.44,1000
654.62,1052.87,998.38,1157.67,981.55,718.48,869.52,1505.57,911.09,775.72,
1118.34,975.42,1132.92,947.14,659.60,1200.15,1033.66,1022.35,1227.38,907.94,
1435.88,1190.44,778.84,1326.98,1162.43,1246.73,1043.58,1034.07,1116.40,934.84,
1126.82,1211.50,932.40,1022.74,1188.63,907.78,1015.64,887.22,1393.42,1067.60,
1224.72,1336.84,1154.76,1238.44,1234.72,1095.73,748.32,1025.56,923.02,1071.70,
1112.84,1200.78,1209.95,1422.25,1245.29,693.02,1112.20,967.55,970.48,766.32,
974.69,849.93,833.45,902.90,954.75,1382.77,1918.39,1162.03,1520.63]

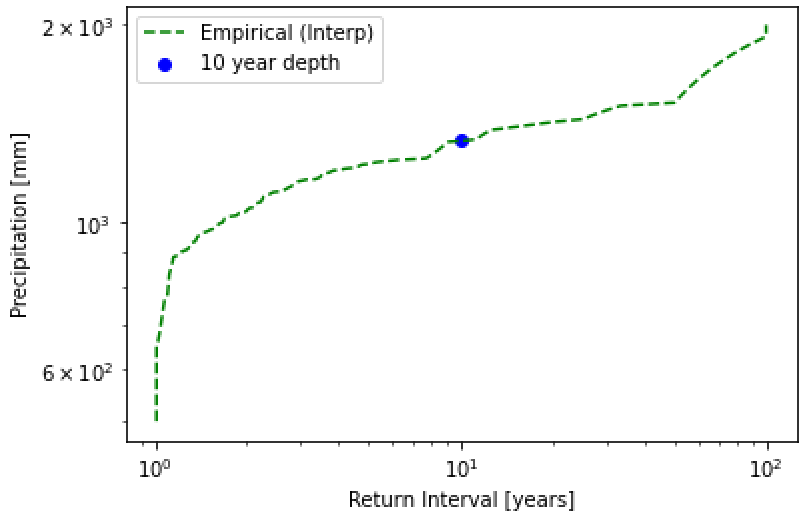
## Methods:

The years and the annual rain data was inputted into a dataframe for ease of manipulation. The dataframe was sorted so that the rainfall data was arranged in ascending order. The ranks were then added to the dataframe making sure that the added array ranged from 1 to 99. A for loop was constructed to calculate the empirical distribution and save it in an empty array F. This array was saved into the dataframe. A range of precipitation values from 500 to 2000 were generated. An interpolation was performed over the range of 500 to 2000 with the sorted annual rain data and the calculatd emipirical distribution. A plot was developed showcasing the interpolated values. Finally the return interval was generated based on the interpolation. The precipitation versus the return interval was plotted, and the 10 year event was estimated by looking within this range.

In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Data
Rain_input = [1021.06,1052.48,1330.55,889.06,1407.20,886.63,1163.28,1168.88,897.59,124
993.65,963.82,1240.87,958.52,1144.11,1017.51,1252.29,1074.10,897.10,1284.55,1097.71,11
654.62,1052.87,998.38,1157.67,981.55,718.48,869.52,1505.57,911.09,775.72,
1118.34,975.42,1132.92,947.14,659.60,1200.15,1033.66,1022.35,1227.38,907.94,
1435.88,1190.44,778.84,1326.98,1162.43,1246.73,1043.58,1034.07,1116.40,934.84,
1126.82,1211.50,932.40,1022.74,1188.63,907.78,1015.64,887.22,1393.42,1067.60,
1224.72,1336.84,1154.76,1238.44,1234.72,1095.73,748.32,1025.56,923.02,1071.70,
1112.84,1200.78,1209.95,1422.25,1245.29,693.02,1112.20,967.55,970.48,766.32,
974.69,849.93,833.45,902.90,954.75,1382.77,1918.39,1162.03,1520.63]

Year_input = np.arange(1901,1999+1,1)
# Created a Dataframe with Years and Associated Data
corvo_rain = {
    "year": Year_input,
    "annual_rain":Rain_input,
}

# Sorted the Data
corvo_df = pd.DataFrame(corvo_rain)

corvo_df_sort = corvo_df.sort_values(by=['annual_rain'], ascending = True)

# Reset the Indicies and then dropped the old index column
corvo_df_sort = corvo_df_sort.reset_index()
corvo_df_sort = corvo_df_sort.drop(columns = ['index'])

# added rank
corvo_df_sort['Rank'] = np.arange(1, len(Year_input)+1)
rank = corvo_df_sort['Rank']
# Estimated the Empirical Distribution of the Rainfall (F(x))

F = np.empty(len(rank))
values = np.arange(len(rank))
for num in values:
    F[num] =(rank[num]/((len(rank))+1))

# Called all of the relevant variables
corvo_df_sort["F(x)"] = F
F_x = corvo_df_sort["F(x)"]
year = corvo_df_sort["year"]
rank = corvo_df_sort["Rank"]
annual_rain = corvo_df_sort["annual_rain"]
#print(corvo_df_sort["F(x)"].head(99))
# Utilized Interp CDF
## First needed to get the x coordinates, and for this got the max and min values for
rainmin, rainmax =annual_rain.min(), annual_rain.max()


## generated x values going from 500 to 2000
Precip_eval = np.arange(500,2000,1)
```

```
## used np.interp to return the linear interpolation for further estimation
pptInterpCDF = np.interp(Precip_eval,annual_rain,F_x)

fig4 = plt.figure()
ax4_1 = fig4.add_subplot(111)
# ax4_1.scatter,F_x, color ="tab:red", label = "observed")
ax4_1.plot(annual_rain,F_x,'r.')
ax4_1.plot(Precip_eval,pptInterpCDF,'g--')
ax4_1.set_xlabel('Rainfall [mm]')
ax4_1.set_ylabel('F(x)')


# Solve for the RI using Interp method
Year = 10
RI_interp = 1/(1-pptInterpCDF)
fig4_2 = plt.figure()
ax4_2 = fig4_2.add_subplot(111)
ax4_2.plot(RI_interp,Precip_eval,'g--',label='Empirical (Interp)')
ax4_2.scatter(10,Precip_eval[np.argmin((RI_interp-Year)**2)], color = 'b', label = '1
ax4_2.set_yscale('log')
ax4_2.set_xscale('log')
ax4_2.set_xlabel("Return Interval [years]")
ax4_2.set_ylabel("Precipitation [mm]")
ax4_2.legend()
plt.show()
print('The %d-year storm has a depth of %.1fmm using an interpolated CDF' % (
    Year, Precip_eval[np.argmin((RI_interp-Year)**2)]))
```





```
The 10-year storm has a depth of 1331.0mm using an interpolated CDF
```

## Results:

The 10 year rain event was determined to have a depth of 1331 mm.

## Discussion:

Using the interpolation method the 10 year rain event was determined to be 1331 mm. This value is the estimated depth of precipitation that would occur over the course of ten years. It is possible that the interpolation method slightly underestimated the value, for this reason it would be prudent for future implementations to run multiple different simulations with larger datasets.

# References

Massachusetts Audubon Society. Groundwater Tour. Retrieved from https://www.oars3rivers.org/river/watercycle/groundwater

Natural Resources Conservation Service. (2009). Watershed Yield.