ATTACHMENT 5

Coded Calculation of Water Consumption

Second Revised Yearly Water Consumption by Cattle at Well 10A-5-W06

June 24, 2016

1 Calculation of Water Consumption by Cattle

1.1 Imports

Import NumPy and matplotlib

```
In [1]: %matplotlib inline

        import numpy
        import matplotlib
        from matplotlib import pylab, mlab, pyplot
        np = numpy
        plt = pyplot
        #import scipy
```

1.2 Raw Data

`DAYS` - tuple of all days from Jan 1 to Dec 31, starting from 0

`TEMPS[day]` - tuple giving ambient daytime temperature in deg F for Jan 1 through Dec 31

`GROWING_CATTLE`, `WINTERING_PREGNANT_COWS`, `LACTATING_COWS`, `MATURE_COWS`, `MATURE_BULLS` - dictionaries of form {weight in lbs: {temperature in F: daily water intake in gal}}

`TIME_GROUPS` - tuple of tuples of days for which certain cattle numbers are known

`n_pcows`, `n_lcows`, etc. - cattle numbers for each time group

`month_to_day()` - splits cattle numbers, in input each index corresponds to a time group, return with each index corresponding to a single day

`nmsu_water_intake()` - New Mexico State University's function for water consumption

`HERD` - dictionary with keys of cattle category and values of tuples of number of that type for each day

`CATS` - dictionary with keys of cattle category and values of tuples of tuples of water consumption by day and cattle numbers by day

```
In [2]: TEMPS = (
            37.3,37.3,37.5,37.5,37.5,36.8,36.8,36.8,37.5,37.5,37.5,37.5,
            37.5,37.5,37.5,37.5,37.5,37.5,37.5,37.5,37.8,37.8,38.5,38.5,
            38.5,38.5,38.5,38.5,39.5,39.5,39.5,39.5,40.3,40.3,40.5,40.5,
            40.5,41.3,41.3,41.3,41.3,42.3,42.3,42.3,42.3,42.3,43.0,43.3,
```

```
    43.3,43.3,43.3,44.0,44.0,44.0,44.0,44.3,44.3,45.0,45.0,45.0,
    45.0,45.0,46.0,46.0,46.0,46.0,46.8,46.8,46.8,46.8,47.0,47.8,
    47.8,47.8,48.5,48.5,48.5,48.8,49.5,49.5,49.5,49.5,49.5,50.3,
    50.3,50.3,51.3,51.3,51.3,51.3,52.0,52.0,52.0,52.8,53.0,53.0,
    53.8,53.8,53.8,53.8,54.8,54.8,54.8,55.5,55.5,55.5,55.8,56.5,
    56.5,56.5,57.3,57.5,57.5,57.5,58.3,58.5,58.5,59.3,59.3,59.5,
    59.5,60.3,60.3,60.5,61.3,61.3,61.3,62.3,62.3,62.3,62.3,63.3,
    63.3,63.3,64.3,64.3,64.3,65.0,65.3,66.0,66.0,66.0,67.0,67.0,
    67.0,67.8,68.0,68.8,68.8,69.0,69.8,69.8,70.5,70.8,70.8,71.5,
    71.5,72.5,72.5,72.5,72.5,73.5,73.5,73.5,74.5,74.5,74.5,74.8,
    74.8,75.5,75.8,75.8,75.8,76.0,76.0,76.0,76.3,76.3,76.3,76.5,
    77.3,78.3,78.3,78.3,78.5,78.5,78.5,78.5,78.8,78.8,78.0,78.0,
    78.3,78.3,78.3,78.3,78.3,78.3,78.5,77.8,77.8,77.8,77.8,77.8,
    77.8,77.0,77.0,77.0,77.0,77.0,77.0,77.0,76.8,76.8,76.0,76.0,
    76.0,76.0,76.0,76.0,75.8,75.0,75.0,75.0,75.0,75.0,75.0,74.8,
    74.8,74.0,74.0,74.0,74.0,73.8,73.8,73.8,73.8,73.0,73.0,72.8,
    72.8,72.8,72.8,72.5,72.5,71.8,71.8,71.5,71.5,71.5,71.5,70.5,
    70.5,70.5,70.5,70.3,69.5,69.5,69.3,69.3,68.5,68.3,68.3,67.5,
    67.3,67.3,67.0,66.3,66.3,66.0,65.3,65.3,65.0,64.3,64.0,64.0,
    63.0,63.0,62.3,62.0,62.0,61.0,61.0,60.3,60.0,60.0,59.3,59.0,
    58.3,58.0,58.0,57.3,57.0,56.3,56.0,55.3,55.3,54.3,54.3,53.5,
    53.3,52.5,52.5,52.3,53.3,52.3,52.3,51.5,51.3,51.3,50.5,50.3,
    49.5,49.5,48.5,48.5,47.8,47.5,47.5,46.8,46.8,45.8,45.8,45.8,
    44.8,44.8,44.0,43.8,43.8,43.0,43.0,43.0,42.0,42.0,41.0,41.0,
    41.0,41.0,40.3,40.3,40.0,40.0,40.0,39.3,39.3,39.3,39.3,39.0,
    39.0,38.3,38.3,38.3,38.3,38.3,38.3,38.3,37.5,37.5,37.5,37.5,
    37.5,37.5,37.5,37.5,37.5
)

DAYS = tuple(i for i in range(len(TEMPS)))
JAN = tuple(i for i in range(0,31))
FEB = tuple(i for i in range(31,59))
MAR = tuple(i for i in range(59,90))
APR = tuple(i for i in range(90,120))
MAY = tuple(i for i in range(120,151))
JUN = tuple(i for i in range(151,181))
JUL = tuple(i for i in range(181,212))
AUG = tuple(i for i in range(212,243))
SEP = tuple(i for i in range(243,273))
OCT = tuple(i for i in range(273,304))
NOV = tuple(i for i in range(304,334))
DEC = tuple(i for i in range(334,365))


GROWING_CATTLE = {
    400: {40: 4.0, 50: 4.3, 60: 5.0, 70: 5.8, 80: 6.7, 90: 9.5},
    600: {40: 5.3, 50: 5.8, 60: 6.6, 70: 7.8, 80: 8.9, 90: 12.7},
    800: {40: 6.3, 50: 6.8, 60: 7.9, 70: 9.2, 80: 10.6, 90: 15.0}
```

```
}

WINTERING_PREGNANT_COWS = {
    1000: {40: 6.0, 50: 6.5, 60: 7.4, 70: 8.7}
}

LACTATING_COWS = {
    1000: {40: 11.4, 50: 12.6, 60: 14.5, 70: 16.9, 80: 17.9, 90: 16.2}
}

MATURE_COWS = {
    1000: {40: 9.0, 50: 9.7, 60: 11.2, 70: 13.1, 80: 15.1, 90: 21.4}
}

MATURE_BULLS = {
    1600: {40: 8.7, 50: 9.4, 60: 10.8, 70: 12.6, 80: 14.5, 90: 20.6}
}

n_pcows = (
    77.36,77.36,58.02,19.44,0.00,5.96,15.82,16.77,17.71,18.62,
    19.56,0.,0.,77.36
)
n_lcows = (
    0.,0.,18.70,56.00,74.42,74.04,184.18,183.23,182.29,181.38,
    180.44,0.,0.,0.
)
n_mcows = (
    2.64,2.64,3.28,4.56,5.20,0.,0.,0.,0.,0.,0.,0.,2.64
)
n_marcv = (
    0.,0.,18.70,18.60,18.51,18.42,45.81,45.57,45.34,45.11,44.88,
    0.,0.,0.
)
n_aprcv = (
    0.,0.,0.,37.40,37.21,37.02,92.09,91.62,91.15,90.69,90.23,0.,
    0.,0.
)
n_maycv = (
    0.,0.,0.,0.,18.70,18.60,46.28,46.04,45.80,45.57,45.34,0.,0.,0.
)
n_bulls = (
    8.,8.,8.,8.,8.,8.,20,20,20,20,20,8.,8.,8.
)

TIME_GROUPS = (
    JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV[:15],
    NOV[15:], DEC[:15], DEC[15:]
)
```

```
def month_to_day(c_type, time_groups=TIME_GROUPS):
    """
    Takes cattle numbers for given time ranges and returns as a
    tuple of cattle numbers per day.
    """
    num = [[c_type[i] for j in time_groups[i]]\
           for i in range(len(time_groups))]
    flat_num = [i for j in num for i in j]
    assert len(flat_num)==365,\
    "month_to_day error: wrong number of days in year"
    return(tuple(flat_num))

def nmsu_water_intake(weight, cat):
    """
    Returns daily water consumption in gal/day for a
    lactating cow or a calf using NMSU method.
    """
    MT = 80
    PP = 0.10
    DS = {
        'l_cows': 0.1,
        'mar_calves': 0.2,
        'apr_calves': 0.2,
        'may_calves': 0.2
    }
    DMI = {
        'l_cows': 0.0225*weight,
        'mar_calves': 0.025*weight,
        'apr_calves': 0.025*weight,
        'may_calves': 0.025*weight
    }
    intake = (-18.67+(0.3937*MT)+(2.432*DMI[cat])-(3.87*PP)\
              -(4.437*DS[cat]))/3.77
    return(intake)

HERD = {
    'pregnant cows': month_to_day(n_pcows),
    'lactating cows': month_to_day(n_lcows),
    'mature cows': month_to_day(n_mcows),
    'march calves': month_to_day(n_marcv),
    'april calves': month_to_day(n_aprcv),
    'may calves': month_to_day(n_maycv),
    'bulls': month_to_day(n_bulls)
}

CATS = {
    'wp_cows': (WINTERING_PREGNANT_COWS, HERD['pregnant cows']),
```

```
        'l_cows': (LACTATING_COWS, HERD['lactating cows']),
        'm_cows': (MATURE_COWS, HERD['mature cows']),
        'mar_calves': (GROWING_CATTLE, HERD['march calves']),
        'apr_calves': (GROWING_CATTLE, HERD['april calves']),
        'may_calves': (GROWING_CATTLE, HERD['may calves']),
        'm_bulls': (MATURE_BULLS, HERD['bulls'])
    }
```

1.3 Calculations for Single Cattle

Calculate the water consumption of any size and class of cattle by getting data from the table via linear interpolation or linear extrapolation. i.e. to get y for a given x, the two closest x-value data points are selected (x_0, y_0) and (x_1, y_1), and y is calculated by

$$y = \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) + y_0.$$

```
In [3]: def calf_age(cat, day, initial_weight=80):
            """Returns the age of the calf."""
            if cat == 'mar_calves':
                age = day - MAR[0]
            elif cat == 'apr_calves':
                age = day - APR[0]
            elif cat == 'may_calves':
                age = day - MAY[0]
            else:
                assert cat=='mar_calves' or cat=='apr_calves' or\
                cat=='may_calves',\
                "Error: non-calf category sent to calf_age()"
            return(age)

        def daily_weight(cat, initial_weight, day):
            """Returns the weight for the given day."""
            if cat!='mar_calves' and cat!='apr_calves' and\
            cat!='may_calves' and cat in CATS:
                return(initial_weight)
            else:
                age = calf_age(cat, day)
            weight = lambda x: (637.5-80)/214*x + 80
            return(weight(age))

        def closest_value_picker(values, n):
            """
            Given a list of values and a number, returns the
            two closest as a list, low number first.
            """
            ordered = sorted(values)
            for v in ordered:
                if v >= n:
```

```
            high = v
            break
        else:
            high = v
    if n == ordered[0] or n == ordered[-1]:
        low = high
        return([low, high])
    elif n > ordered[0]:
        low = ordered[ordered.index(high) - 1]
        assert high > low,\
        "chart weight selection error occurred"
        return([low, high])
    else:
        low = high
        high = ordered[ordered.index(low) + 1]
        assert high > low,\
        "chart weight selection error occurred"
        return([low, high])

def cons(cat, initial_weight, day):
    """
    Returns water consumption for a given day and a
    given size and class of cattle.
    """
    assert cat in CATS, "category not in CATS"
    w = daily_weight(cat, initial_weight, day)
    t = TEMPS[day]
    closest_weights = closest_value_picker(CATS[cat][0], w)
    closest_temps = \
    closest_value_picker(CATS[cat][0][closest_weights[0]], t)
    t_0 = closest_temps[0]
    t_1 = closest_temps[1]
    if t_0 != t_1: # interpolate for temperature
        c_0l = CATS[cat][0][closest_weights[0]][t_0]
        c_1l = CATS[cat][0][closest_weights[0]][t_1]
        c_l = (c_1l - c_0l) / (t_1 - t_0) * (t - t_0) + c_0l
        c_0h = CATS[cat][0][closest_weights[1]][t_0]
        c_1h = CATS[cat][0][closest_weights[1]][t_1]
        c_h = (c_1h - c_0h) / (t_1 - t_0) * (t - t_0) + c_0h
    else:
        c_l = CATS[cat][0][closest_weights[0]][t_0]
        c_h = CATS[cat][0][closest_weights[1]][t_0]
    w_0 = closest_weights[0]
    w_1 = closest_weights[1]
    if w_0 != w_1: # interpolate for weight
        c = (c_h - c_l) / (w_1 - w_0) * (w - w_0) + c_l
    else:
        c = c_l
```

```
    if cat=='mar_calves' and calf_age(cat, day)<92 and\
    calf_age(cat, day)>=0:
        first_three_months = np.linspace(0, c, 92)
        return(first_three_months[calf_age(cat, day)])
    elif cat=='apr_calves' and calf_age(cat, day)<91 and\
    calf_age(cat, day)>=0:
        first_three_months = np.linspace(0, c, 91)
        return(first_three_months[calf_age(cat, day)])
    elif cat=='may_calves' and calf_age(cat, day)<92 and\
    calf_age(cat, day)>=0:
        first_three_months = np.linspace(0, c, 92)
        return(first_three_months[calf_age(cat, day)])
    else:
        return(c)
```

1.4 Calculations for Groups of Cattle

```
In [4]: def daily_consumption(day, method='nrc'):
            """
            Returns how much water is consumed on the given day for the herd.
            """
            wp_cows_cons = cons('wp_cows', 1000, day) *\
            CATS['wp_cows'][1][day]
            m_cows_cons = cons('m_cows', 1000, day) *\
            CATS['m_cows'][1][day]
            m_bulls_cons = cons('m_bulls', 1600, day) *\
            CATS['m_bulls'][1][day]
            if method=='nrc':
                l_cows_cons = cons('l_cows', 1000, day) *\
                CATS['l_cows'][1][day]
                mar_calves_cons = cons('mar_calves', 80, day) *\
                CATS['mar_calves'][1][day]
                apr_calves_cons = cons('apr_calves', 80, day) *\
                CATS['apr_calves'][1][day]
                may_calves_cons = cons('may_calves', 80, day) *\
                CATS['may_calves'][1][day]
            elif method=='nmsu':
                l_cows_cons = nmsu_water_intake(1000, 'l_cows') *\
                CATS['l_cows'][1][day]
                mar_calves_cons = nmsu_water_intake(daily_weight(\
                'mar_calves', 80, day), 'mar_calves')\
                * CATS['mar_calves'][1][day]
                apr_calves_cons = nmsu_water_intake(daily_weight(\
                'apr_calves', 80, day), 'apr_calves')\
                * CATS['apr_calves'][1][day]
                may_calves_cons = nmsu_water_intake(daily_weight(\
                'may_calves', 80, day), 'may_calves')\
                * CATS['may_calves'][1][day]
```

```
    else:
        print("Error: wrong method passes do daily_consumption()")
    total = wp_cows_cons+l_cows_cons+m_cows_cons+\
    mar_calves_cons+apr_calves_cons+may_calves_cons+m_bulls_cons
    return(total)

def total_consumption(start, end, method='nrc'):
    """
    Returns YTD water consumption for the given day for the herd.
    """
    total = 0
    for i in range(start, end+1):
        total += daily_consumption(i, method)
    return(total)

def daily_cowcalf_consumption(day, method='nrc'):
    """
    Returns how much water is consumed on the given day for all
    cow-calf pairs.
    """
    if method=='nrc':
        l_cows_cons = cons('l_cows', 1000, day)\
        * CATS['l_cows'][1][day]
        mar_calves_cons = cons('mar_calves', 80, day)\
        * CATS['mar_calves'][1][day]
        apr_calves_cons = cons('apr_calves', 80, day)\
        * CATS['apr_calves'][1][day]
        may_calves_cons = cons('may_calves', 80, day)\
        * CATS['may_calves'][1][day]
    elif method=='nmsu':
        l_cows_cons = nmsu_water_intake(1000, 'l_cows')\
        * CATS['l_cows'][1][day]
        mar_calves_cons = nmsu_water_intake(daily_weight(\
        'mar_calves', 80, day), 'mar_calves')\
        * CATS['mar_calves'][1][day]
        apr_calves_cons = nmsu_water_intake(daily_weight(\
        'apr_calves', 80, day), 'apr_calves')\
        * CATS['apr_calves'][1][day]
        may_calves_cons = nmsu_water_intake(daily_weight(\
        'may_calves', 80, day), 'may_calves')\
        * CATS['may_calves'][1][day]
    else:
        print("Error: wrong method passes do daily_consumption()")
    total = l_cows_cons+mar_calves_cons+apr_calves_cons\
    +may_calves_cons
    return(total)

def total_cowcalf_consumption(start, end, method='nrc'):
```

```
    """
    Returns YTD water consumption for the given day for all
    cow-calf pairs.
    """
    total = 0
    for i in range(start, end+1):
        total += daily_cowcalf_consumption(i, method)
    return(total)
```

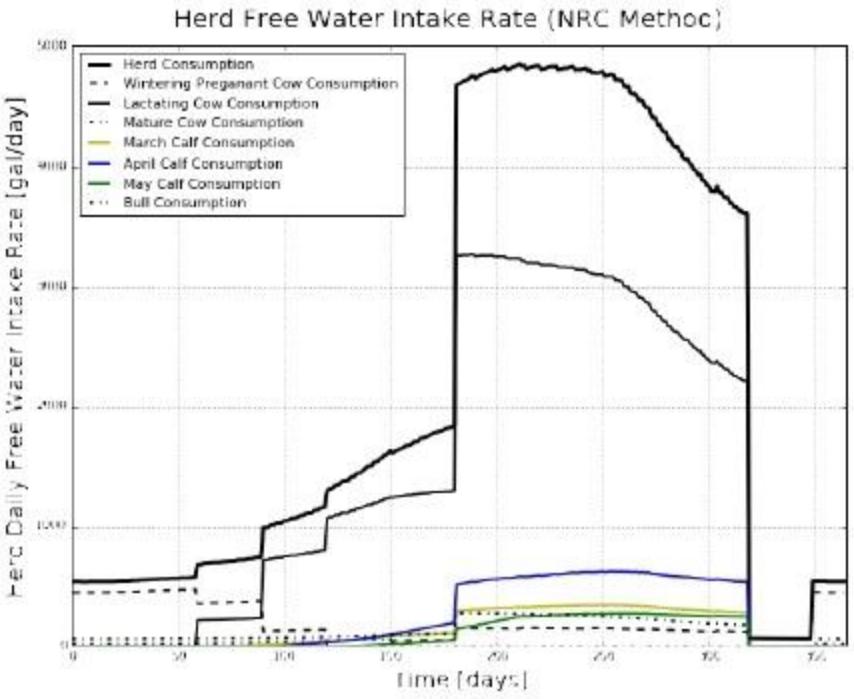
1.5 Herd Free Water Intake Rate NRC Method

Creates a graph of the free water intake rate of each class of cattle and the total herd as a function of day.

```
In [5]: x = list(DAYS)
        t = [daily_consumption(i) for i in x]
        c_wp = [cons('wp_cows', 1000, day) * CATS['wp_cows'][1][day] for\
                day in x]
        c_l = [cons('l_cows', 1000, day) * CATS['l_cows'][1][day] for\
               day in x]
        c_m = [cons('m_cows', 1000, day) * CATS['m_cows'][1][day] for\
               day in x]
        cv_mar = [cons('mar_calves', 80, day) * CATS['mar_calves'][1][day]\
                  for day in x]
        cv_apr = [cons('apr_calves', 80, day) * CATS['apr_calves'][1][day]\
                  for day in x]
        cv_may = [cons('may_calves', 80, day) * CATS['may_calves'][1][day]\
                  for day in x]
        b = [cons('m_bulls', 1600, day) * CATS['m_bulls'][1][day] \
             for day in x]

        # create plot
        fig1, ax1 = plt.subplots()
        fig1.suptitle('Herd Free Water Intake Rate (NRC Method)',\
                      fontsize=20, y=0.95)
        lns1 = ax1.plot(x, t, 'k-', lw=3, label='Herd Consumption')
        lns2 = ax1.plot(x, c_wp, 'k--', lw=2, label=\
                        'Wintering Preganant Cow Consumption')
        lns3 = ax1.plot(x, c_l, 'k-', lw=2, label=\
                        'Lactating Cow Consumption')
        lns4 = ax1.plot(x, c_m, 'k-.', lw=2, label=\
                        'Mature Cow Consumption')
        lns5 = ax1.plot(x, cv_mar, 'y-', lw=2, label=\
                        'March Calf Consumption')
        lns6 = ax1.plot(x, cv_apr, 'b-', lw=2, label=\
                        'April Calf Consumption')
        lns7 = ax1.plot(x, cv_may, 'g-', lw=2, label=\
                        'May Calf Consumption')
```

```
lns8 = ax1.plot(x, b, 'k-.', lw=2, label=\
                'Bull Consumption')

# edit axis 1
ax1.grid(b=True, which='major', axis='both')
ax1.set_xscale('linear')
ax1.set_yscale('linear')
ax1.set_xlim([0, 365])
ax1.set_xlabel('Time [days]', fontsize=18)
ax1.set_ylabel(\
    'Herd Daily Free Water Intake Rate [gal/day]',\
                fontsize=18)
ax1.legend(fontsize = 11, loc="upper left")

# display plot
fig1.set_size_inches(11,8.5)
plt.show

print("Average consumption rate for the herd is \n \
{0:.0f} gallons per day.".format(np.average(t)))
print("The max daily consumption for the herd is \n \
{0:.0f} gallons per day.".format(max(t)))
print("The min daily consumption for the herd is \n \
{0:.0f} gallons per day.".format(min(t)))
```

```
Average consumption rate for the herd is
 2235 gallons per day.
The max daily consumption for the herd is
 4848 gallons per day.
The min daily consumption for the herd is
 69 gallons per day.
```



1.6 Cow-Calf Pair Free Water Intake Rate NRC Method

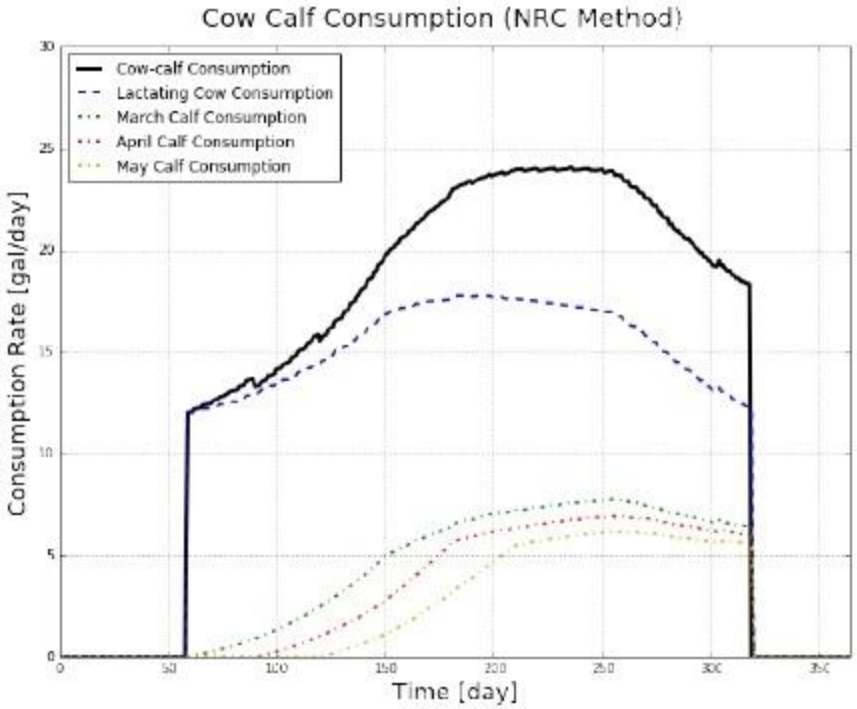
Creates a graph of the free water intake rate for the average (using as weights the relative number in the age group) cow-calf pair, a cow and a calf as a function of day.

```
In [6]: x = list(DAYS)
        c_l = [cons('l_cows', 1000, day)\
               * (day in range(MAR[0],NOV[15]+1)) for day in x]
        cv_mar = [cons('mar_calves', 80, day)\
                  * (day in range(MAR[0],NOV[15]+1)) for day in x]
        cv_apr = [cons('apr_calves', 80, day)\
                  * (day in range(APR[0],NOV[15]+1)) for day in x]
        cv_may = [cons('may_calves', 80, day)\
                  * (day in range(MAY[0],NOV[15]+1)) for day in x]

        c_l_tot = [cons('l_cows', 1000, day)\
                   * CATS['l_cows'][1][day] for day in x]
        cv_mar_tot = [cons('mar_calves', 80, day)\
                      * CATS['mar_calves'][1][day]\
```

```
                * (day in range(MAR[0],NOV[15]+1)) for day in x]
cv_apr_tot = [cons('apr_calves', 80, day)\
                * CATS['apr_calves'][1][day]\
                * (day in range(APR[0],NOV[15]+1)) for day in x]
cv_may_tot = [cons('may_calves', 80, day)\
                * CATS['may_calves'][1][day]\
                * (day in range(MAY[0],NOV[15]+1)) for day in x]

t = []
mar_count = [CATS['mar_calves'][1][day]\
                * (day in range(MAR[0],NOV[15]+1)) for day in x]
apr_count = [CATS['apr_calves'][1][day]\
                * (day in range(APR[0],NOV[15]+1)) for day in x]
may_count = [CATS['may_calves'][1][day]\
                * (day in range(MAY[0],NOV[15]+1)) for day in x]
count = [mar_count[i]+apr_count[i]+may_count[i] for i in x]
for day in x:
    if CATS['l_cows'][1][day] == 0:
        t += [0]
    else:
        t += [(c_l_tot[day]+cv_mar_tot[day]+\
               cv_apr_tot[day]+cv_may_tot[day])/count[day]]

# create plot
fig1, ax1 = plt.subplots()
fig1.suptitle('Cow Calf Consumption (NRC Method)',\
              fontsize=20, y=0.95)
lns1 = ax1.plot(x, t, 'k-', lw=3,\
                label='Cow-calf Consumption')
lns2 = ax1.plot(x, c_l, 'b--', lw=2,\
                label='Lactating Cow Consumption')
lns3 = ax1.plot(x, cv_mar, 'g-.', lw=2,\
                label='March Calf Consumption')
lns4 = ax1.plot(x, cv_apr, 'r-.', lw=2,\
                label='April Calf Consumption')
lns5 = ax1.plot(x, cv_may, 'y-.', lw=2,\
                label='May Calf Consumption')

# edit axis 1
ax1.grid(b=True, which='major', axis='both')
ax1.set_xscale('linear')
ax1.set_yscale('linear')
ax1.set_xlim([0, DEC[-1]])
ax1.set_ylim([0, 30])
ax1.set_xlabel('Time [day]', fontsize=18)
ax1.set_ylabel('Consumption Rate [gal/day]', fontsize=18)
ax1.legend(loc="upper left")
```

```
# display plot
fig1.set_size_inches(11,8.5)
plt.show

t = t[MAR[0]:NOV[14]]
print("Average for a cow-calf pair is \n\
{0:.2f} gallons per day.".format(np.average(t)))
print("The max daily consumption for a cow-calf pair is \n\
{0:.2f} gallons per day.".format(max(t)))
print("The min daily consumption for a cow-calf pair is \n\
{0:.2f} gallons per day.".format(min(t)))
```

```
Average for a cow-calf pair is
19.66 gallons per day.
The max daily consumption for a cow-calf pair is
24.05 gallons per day.
The min daily consumption for a cow-calf pair is
12.00 gallons per day.
```



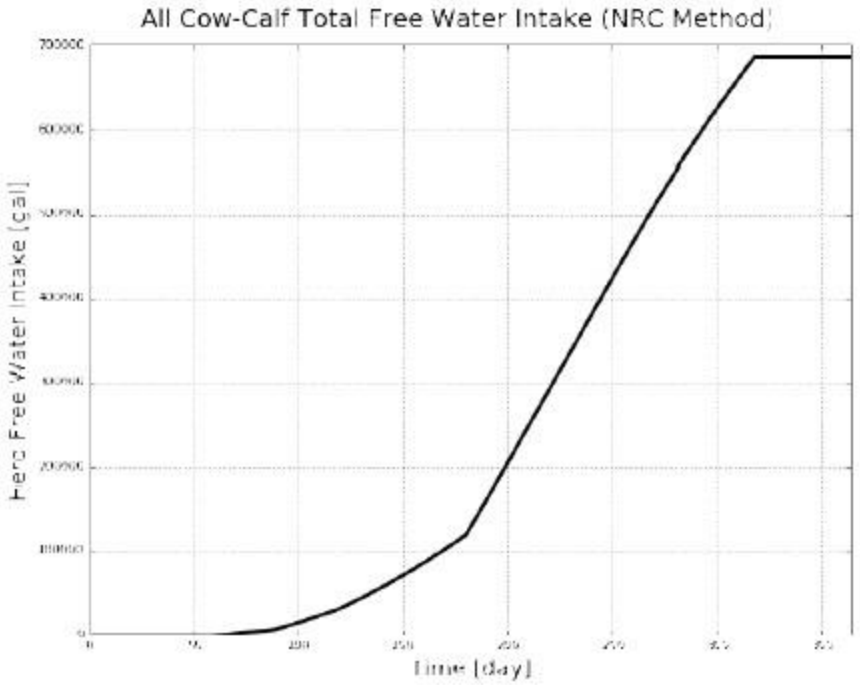
13

1.7 Cow-Calf Total Free Water Intake NRC Method

Creates a graph of the free water intake of all cow-calf pairs as a function of day.

```
In [7]: x = list(DAYS)
        y = [total_cowcalf_consumption(0, i) for i in x]

        # create plot
        fig2, ax2 = plt.subplots()
        fig2.suptitle(
            'All Cow-Calf Total Free Water Intake (NRC Method)',\
            fontsize=20, y=0.95)
        ax2.plot(x, y, 'k-', lw=3)

        # edit axis 1
        ax2.grid(b=True, which='major', axis='both')
        ax2.set_xscale('linear')
        ax2.set_yscale('linear')
        ax2.set_xlim(0, DEC[-1])
        ax2.set_xlabel('Time [day]', fontsize=18)
        ax2.set_ylabel('Herd Free Water Intake [gal]', fontsize=18)

        # display plot
        fig2.set_size_inches(11,8.5)
        plt.show

        print("The total yearly consumption for all cow calf \n\
        pairs is {0:.0f} gallons.".format(max(y)))
        winter_season = total_cowcalf_consumption(0, JUN[-1]) +\
        total_cowcalf_consumption(NOV[15], DEC[-1])
        print("The winter season consumption for all cow calf pairs, \n\
        November 16 through June 30, is {0:.0f} gallons."\
              .format(winter_season))
        summer_season = total_cowcalf_consumption(JUL[0], NOV[14])
        print("The summer season consumption for all cow calf pairs, \n\
        July 1 through November 15, is {0:.0f} gallons."\
              .format(summer_season))
```

```
The total yearly consumption for all cow calf
pairs is 685254 gallons.
The winter season consumption for all cow calf pairs,
November 16 through June 30, is 119489 gallons.
The summer season consumption for all cow calf pairs,
July 1 through November 15, is 565765 gallons.
```



1.8 Herd Total Free Water Intake NRC Method

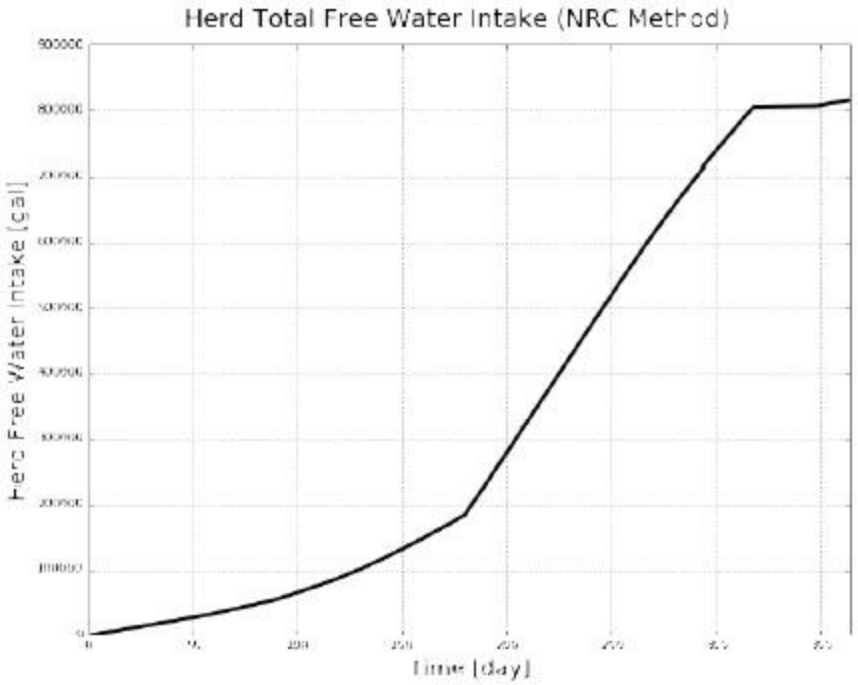
Creates a graph of the free water intake of the herd as a function of day.

```
In [8]: x = list(DAYS)
        y = [total_consumption(0, i) for i in x]

        # create plot
        fig2, ax2 = plt.subplots()
        fig2.suptitle('Herd Total Free Water Intake (NRC Method)',\
                      fontsize=20, y=0.95)
        ax2.plot(x, y, 'k-', lw=3)

        # edit axis 1
        ax2.grid(b=True, which='major', axis='both')
        ax2.set_xscale('linear')
        ax2.set_yscale('linear')
        ax2.set_xlim(0, DEC[-1])
        ax2.set_xlabel('Time [day]', fontsize=18)
```

15

```
ax2.set_ylabel('Herd Free Water Intake [gal]', fontsize=18)

# display plot
fig2.set_size_inches(11,8.5)
plt.show

print("The total yearly consumption for herd is \n\
{0:.0f} gallons.".format(max(y)))
winter_season = total_consumption(0, JUN[-1]) + \
total_consumption(NOV[15], DEC[-1])
print("The winter season consumption for herd, \n\
November 16 through June 30, is {0:.0f} gallons.".\
      format(winter_season))
summer_season = total_consumption(JUL[0], NOV[14])
print("The summer season consumption for herd, July \n\
1 through November 15, is {0:.0f} gallons."\
      .format(summer_season))
```

```
The total yearly consumption for herd is
815802 gallons.
The winter season consumption for herd,
November 16 through June 30, is 195755 gallons.
The summer season consumption for herd, July
1 through November 15, is 620047 gallons.
```



1.9 Herd Free Water Intake Rate NMSU Method

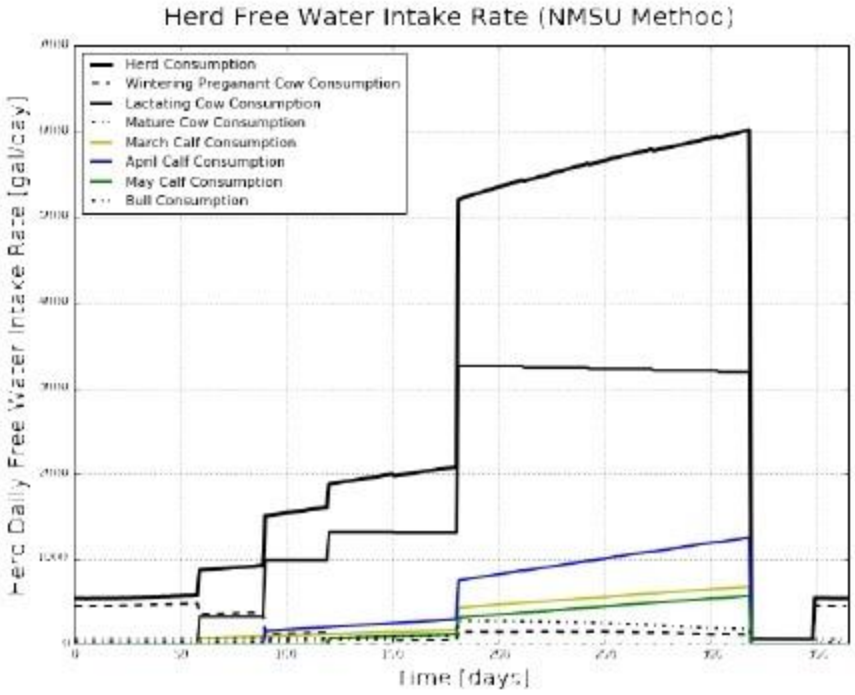
Creates a graph of the free water intake rate of each class of cattle and the total herd as a function of day.

```
In [9]: x = list(DAYS)
        t = [daily_consumption(i, 'nmsu') for i in x]
        c_wp = [cons('wp_cows', 1000, day) \
                * CATS['wp_cows'][1][day] for day in x]
        c_l = [nmsu_water_intake(1000, 'l_cows') \
               * CATS['l_cows'][1][day] for day in x]
        c_m = [cons('m_cows', 1000, day) \
               * CATS['m_cows'][1][day] for day in x]
        cv_mar = [nmsu_water_intake(daily_weight(\
                 'mar_calves', 80, day), 'mar_calves')\
                 * CATS['mar_calves'][1][day] for day in x]
        cv_apr = [nmsu_water_intake(daily_weight(\
                 'apr_calves', 80, day), 'apr_calves')\
                 * CATS['apr_calves'][1][day] for day in x]
```

```
cv_may = [nmsu_water_intake(daily_weight(\
        'may_calves', 80, day), 'may_calves')\
        * CATS['may_calves'][1][day] for day in x]
b = [cons('m_bulls', 1600, day) \
     * CATS['m_bulls'][1][day] for day in x]

# create plot
fig1, ax1 = plt.subplots()
fig1.suptitle(\
    'Herd Free Water Intake Rate (NMSU Method)', \
              fontsize=20, y=0.95)
lns1 = ax1.plot(x, t, 'k-', lw=3, \
                label='Herd Consumption')
lns2 = ax1.plot(x, c_wp, 'k--', lw=2, \
                label='Wintering Preganant Cow Consumption')
lns3 = ax1.plot(x, c_l, 'k-', lw=2, \
                label='Lactating Cow Consumption')
lns4 = ax1.plot(x, c_m, 'k-.', lw=2, \
                label='Mature Cow Consumption')
lns5 = ax1.plot(x, cv_mar, 'y-', lw=2, \
                label='March Calf Consumption')
lns6 = ax1.plot(x, cv_apr, 'b-', lw=2, \
                label='April Calf Consumption')
lns7 = ax1.plot(x, cv_may, 'g-', lw=2, \
                label='May Calf Consumption')
lns8 = ax1.plot(x, b, 'k-.', lw=2, \
                label='Bull Consumption')

# edit axis 1
ax1.grid(b=True, which='major', axis='both')
ax1.set_xscale('linear')
ax1.set_yscale('linear')
ax1.set_xlim([0, 365])
ax1.set_xlabel('Time [days]', fontsize=18)
ax1.set_ylabel(\
        'Herd Daily Free Water Intake Rate [gal/day]', \
              fontsize=18)
ax1.legend(fontsize = 11, loc="upper left")

# display plot
fig1.set_size_inches(11,8.5)
plt.show

print("Average consumption rate for the herd is \n\
{0:.0f} gallons per day.".format(np.average(t)))
print("The max daily consumption for the herd is \n\
{0:.0f} gallons per day.".format(max(t)))
print("The min daily consumption for the herd is \n\
```

```
        {0:.0f} gallons per day.".format(min(t)))
```

```
Average consumption rate for the herd is
2789 gallons per day.
The max daily consumption for the herd is
6018 gallons per day.
The min daily consumption for the herd is
69 gallons per day.
```



1.10 Cow-Calf Pair Free Water Intake Rate NMSU Method

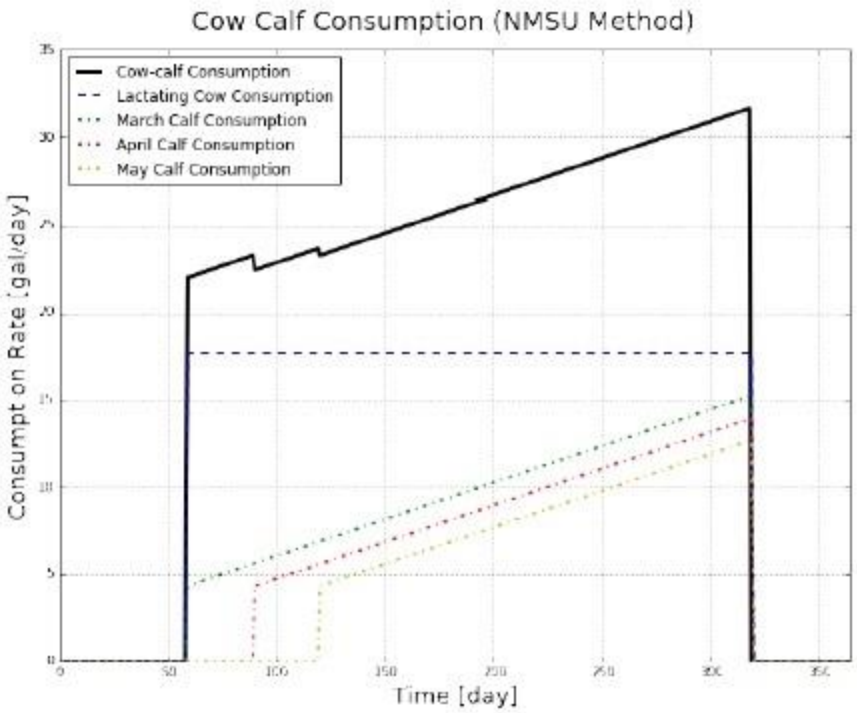
Creates a graph of the free water intake rate for the average (using as weights the relative number in the age group) cow-calf pair, a cow and a calf as a function of day.

```
In [10]: x = list(DAYS)
         c_l = [nmsu_water_intake(1000, 'l_cows') \
                * (day in range(MAR[0],NOV[15]+1)) for day in x]
         cv_mar = [nmsu_water_intake(daily_weight(\
```

```
        'mar_calves', 80, day), 'mar_calves')\
        * (day in range(MAR[0],NOV[15]+1)) for day in x]
cv_apr = [nmsu_water_intake(daily_weight(\
        'apr_calves', 80, day), 'apr_calves')\
        * (day in range(APR[0],NOV[15]+1)) for day in x]
cv_may = [nmsu_water_intake(daily_weight(\
        'may_calves', 80, day), 'may_calves')\
        * (day in range(MAY[0],NOV[15]+1)) for day in x]

c_l_tot = [nmsu_water_intake(1000, 'l_cows') \
          * CATS['l_cows'][1][day] for day in x]
cv_mar_tot = [nmsu_water_intake(daily_weight(\
           'mar_calves', 80, day), 'mar_calves')\
           * CATS['mar_calves'][1][day] \
             * (day in range(MAR[0],NOV[15]+1)) for day in x]
cv_apr_tot = [nmsu_water_intake(daily_weight(\
           'apr_calves', 80, day), 'apr_calves')\
           * CATS['apr_calves'][1][day] \
           * (day in range(APR[0],NOV[15]+1)) for day in x]
cv_may_tot = [nmsu_water_intake(daily_weight(\
           'may_calves', 80, day), 'may_calves')\
           * CATS['may_calves'][1][day]\
             * (day in range(MAY[0],NOV[15]+1)) for day in x]

t = []
mar_count = [CATS['mar_calves'][1][day] \
           * (day in range(MAR[0],NOV[15]+1)) for day in x]
apr_count = [CATS['apr_calves'][1][day] \
           * (day in range(APR[0],NOV[15]+1)) for day in x]
may_count = [CATS['may_calves'][1][day] \
           * (day in range(MAY[0],NOV[15]+1)) for day in x]
count = [mar_count[i]+apr_count[i]+may_count[i] for i in x]
for day in x:
    if CATS['l_cows'][1][day] == 0:
        t += [0]
    else:
        t += [(c_l_tot[day]+cv_mar_tot[day]+\
              cv_apr_tot[day]+cv_may_tot[day])/count[day]]

# create plot
fig1, ax1 = plt.subplots()
fig1.suptitle('Cow Calf Consumption (NMSU Method)', \
             fontsize=20, y=0.95)
lns1 = ax1.plot(x, t, 'k-', lw=3, \
              label='Cow-calf Consumption')
lns2 = ax1.plot(x, c_l, 'b--', lw=2, \
              label='Lactating Cow Consumption')
lns3 = ax1.plot(x, cv_mar, 'g-.', lw=2, \
```

```
                    label='March Calf Consumption')
lns4 = ax1.plot(x, cv_apr, 'r-.', lw=2, \
                    label='April Calf Consumption')
lns5 = ax1.plot(x, cv_may, 'y-.', lw=2, \
                    label='May Calf Consumption')

# edit axis 1
ax1.grid(b=True, which='major', axis='both')
ax1.set_xscale('linear')
ax1.set_yscale('linear')
ax1.set_xlim([0, DEC[-1]])
ax1.set_ylim([0, 35])
ax1.set_xlabel('Time [day]', fontsize=18)
ax1.set_ylabel('Consumption Rate [gal/day]', \
                   fontsize=18)
ax1.legend(loc="upper left")

# display plot
fig1.set_size_inches(11,8.5)
plt.show

t = t[MAR[0]:NOV[14]]
print("Average for a cow-calf pair is {0:.2f} \n\
gallons per day.".format(np.average(t)))
print("The max daily consumption for a cow-calf \n\
pair is {0:.2f} gallons per day.".format(max(t)))
print("The min daily consumption for a cow-calf \n\
pair is {0:.2f} gallons per day.".format(min(t)))
```

```
Average for a cow-calf pair is 26.38
gallons per day.
The max daily consumption for a cow-calf
pair is 31.59 gallons per day.
The min daily consumption for a cow-calf
pair is 22.05 gallons per day.
```



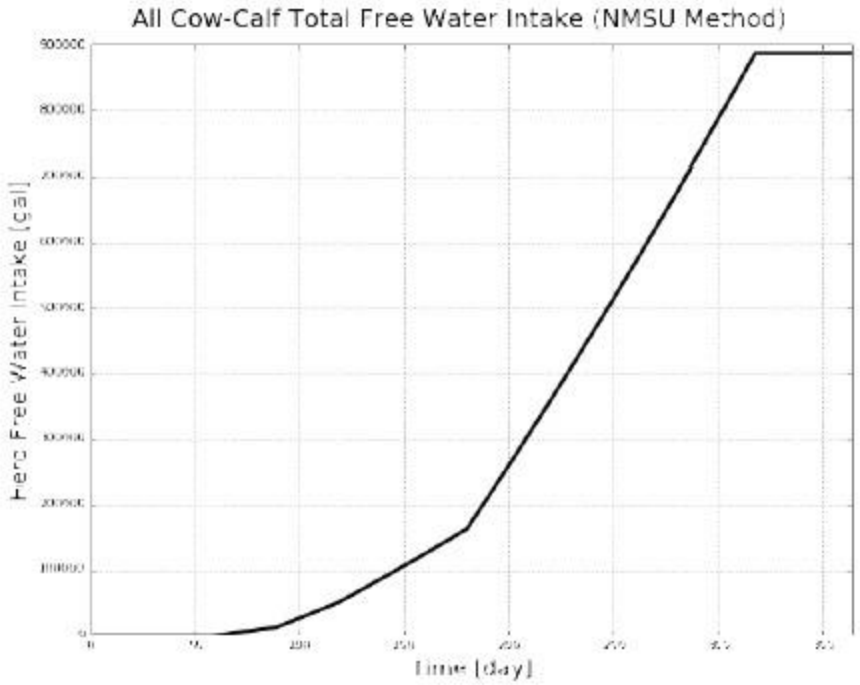
1.11 All Cow-Calf Total Free Water Intake NMSU Method

Creates a graph of the free water intake of all cow-calf pairs as a function of day.

```
In [11]: x = list(DAYS)
         y = [total_cowcalf_consumption(0, i, method='nmsu') \
              for i in x]

         # create plot
         fig2, ax2 = plt.subplots()
         fig2.suptitle('All Cow-Calf Total Free Water Intake \
         (NMSU Method)', fontsize=20, y=0.95)
         ax2.plot(x, y, 'k-', lw=3)

         # edit axis 1
         ax2.grid(b=True, which='major', axis='both')
         ax2.set_xscale('linear')
         ax2.set_yscale('linear')
```

```
ax2.set_xlim(0, DEC[-1])
ax2.set_xlabel('Time [day]', fontsize=18)
ax2.set_ylabel('Herd Free Water Intake [gal]', \
               fontsize=18)

# display plot
fig2.set_size_inches(11,8.5)
plt.show

print("The total yearly consumption for all cow calf \n\
pairs is {0:.0f} gallons.".format(max(y)))
winter_season = total_cowcalf_consumption(0, JUN[-1], \
                method='nmsu') + total_cowcalf_consumption(\
                NOV[15], DEC[-1], method='nmsu')
print("The winter season consumption for all cow \n\
calf pairs, November 16 through June 30, is {0:.0f} gallons."\
      .format(winter_season))
summer_season = total_cowcalf_consumption(JUL[0], NOV[14], \
                                          method='nmsu')
print("The summer season consumption for all cow calf pairs, \n\
July 1 through November 15, is {0:.0f} gallons."\
      .format(summer_season))
```

```
The total yearly consumption for all cow calf
pairs is 887334 gallons.
The winter season consumption for all cow
calf pairs, November 16 through June 30, is 163234 gallons.
The summer season consumption for all cow calf pairs,
July 1 through November 15, is 724100 gallons.
```



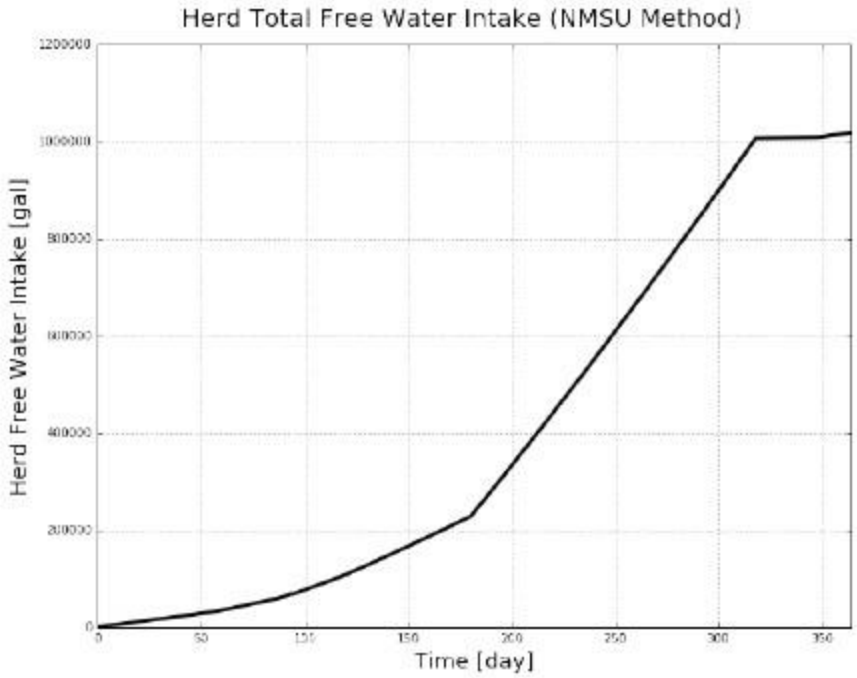
1.12 Herd Total Free Water Intake NMSU Method

Creates a graph of the free water intake of the herd as a function of day.

```
In [12]: x = list(DAYS)
         y = [total_consumption(0, i, method='nmsu') \
              for i in x]

         # create plot
         fig2, ax2 = plt.subplots()
         fig2.suptitle('Herd Total Free Water Intake \
         (NMSU Method)', fontsize=20, y=0.95)
         ax2.plot(x, y, 'k-', lw=3)

         # edit axis 1
         ax2.grid(b=True, which='major', axis='both')
         ax2.set_xscale('linear')
         ax2.set_yscale('linear')
         ax2.set_xlim(0, DEC[-1])
```

```
ax2.set_xlabel('Time [day]', fontsize=18)
ax2.set_ylabel('Herd Free Water Intake [gal]', \
               fontsize=18)

# display plot
fig2.set_size_inches(11,8.5)
plt.show

print("The total yearly consumption for the herd \n\
is {0:.0f} gallons.".format(max(y)))
winter_season = total_consumption(0, JUN[-1], method='nmsu') \
+ total_consumption(NOV[15], DEC[-1], method='nmsu')
print("The winter season consumption for the herd, \n\
November 16 through June 30, is {0:.0f} gallons."\
      .format(winter_season))
summer_season = total_consumption(JUL[0], NOV[14], method='nmsu')
print("The summer season consumption for the herd, \n\
July 1 through November 15, is {0:.0f} gallons."\
      .format(summer_season))
```

```
The total yearly consumption for the herd
is 1017881 gallons.
The winter season consumption for the herd,
November 16 through June 30, is 239499 gallons.
The summer season consumption for the herd,
July 1 through November 15, is 778382 gallons.
```


Herd Total Free Water Intake (NMSU Method)
Herd Free Water Intake [gal]
1200000
1000000
800000
600000
400000
200000
0
0
50
100
150
200
250
300
350
Time [day]

ATTACHMENT 6

Exhibits

The expert report contains exhibits, including documents, photographs, figures and tables, that summarize or support the opinions rendered and which may be used at trial. Trial exhibits will be taken directly from the report and references without alteration of content. Format changes, such as to size and orientation, may be made to facilitate presentation using Microsoft PowerPoint. Bullet lists, summarizing the bases and reasons for these opinions as well as the facts or data that were considered, may be compiled to assist the trier of fact to understand the evidence or to determine a fact in issue.

ATTACHMENT 7

Resume of Craig L. Fredrickson

CRAIG L. FREDRICKSON

Experience Summary

Mr. Fredrickson has diverse professional and management experience derived from a 27-year career within the commercial and federal government nuclear and hazardous waste management industries. His technical expertise includes the assessment of public health and environmental risks posed by chemical and radiotoxic materials, nuclear facility safety analysis, and application of regulations pursuant to the Atomic Energy Act, Resource Conservation and Recovery Act (RCRA), Comprehensive Environmental Response, Compensation, and Liability Act, Clean Air Act, the Nuclear Waste Policy Act, and the National Environmental Policy Act (NEPA). Mr. Fredrickson directed several multi-million-dollar subcontracts to U.S. Department of Energy (DOE) prime contractors, and managed project staffs and organizations composed of up to 80 engineers and scientists. He was a founder and managing director of Benchmark Environmental Corporation. Mr. Fredrickson retired with Benchmark in 2000.

Education and Selected Training

B.S., Nuclear Engineering, Pennsylvania State University, University Park, Pennsylvania, 1973

Basic Management Program, University of New Mexico Graduate School of Management, Albuquerque, New Mexico, 1993

Site Remediation (40-Hour Occupational Safety and Health Administration [OSHA] training), International Technology (IT) Corporation, Albuquerque, New Mexico, 1988

Project Management Professional Development Course, IT Corporation, Albuquerque, New Mexico, 1986

Professional Experience

Benchmark Environmental Corporation
Albuquerque, New Mexico
1989–2000

As a managing director, Mr. Fredrickson was responsible for program management, business development, office operations, and other functions necessary to support an environmental services consulting firm. He served as the president of the company, as its financial officer, and developed and maintained corporate infrastructure including corporate accounting in accordance with the federal acquisition regulation, quality assurance in accordance with nuclear standards, and development of physical facilities. He was one of the founders of the firm.

Mr. Fredrickson's technical expertise is in nuclear facility safety, radioactive mixed waste risk assessment, and regulatory compliance issues. As a consulting engineer, Mr. Fredrickson assisted in conducting nuclear facility safety analysis of the Waste Isolation Pilot Plant (WIPP). He conducted radiological risk analyses and public health evaluations for the WIPP *Final Supplemental Environmental Impact Statement*. He supported the development of the RCRA Part B permit application for the WIPP and was project manager responsible for preparing a National Emission Standards for Hazardous Air Pollutants' quality assurance project plan for the WIPP radioactive emission monitoring and sampling system.

Mr. Fredrickson managed several task order agreements to provide technical support services to Los Alamos National Laboratory (LANL). Mr. Fredrickson directed the preparation of safety analysis reports (SARs) for various LANL facilities including a project to retrieve transuranic (TRU) waste from earthen-covered storage, a TRU waste characterization and transport loading facility, a radioactive waste characterization, reduction, and repackaging facility, and a TRU waste storage and low-level radioactive waste (LLW) disposal facility.

Mr. Fredrickson was project manager and conducted radiological dose modeling for a proposed commercial LLW disposal facility in the mid-qawest. He directed various studies for DOE's Hanford Reservation site, and was the project manager of a subcontract to provide radioactive waste management support services to the Idaho National Engineering Laboratory (INEL).

International Technology (IT) Corporation
Albuquerque, New Mexico
1984–1989

As business unit manager, Mr. Fredrickson was responsible for business development, project management, and technical oversight and supervision of the Environmental Compliance and Assessments business unit at IT's Albuquerque Office. He directed a staff of up to 20 engineers and scientists, managed six major subcontracts with DOE prime contractors, and performed technical assignments related to the management of radioactive and hazardous waste. As program director, he managed a master services agreement with EG&G Idaho, Inc. to provide industrial/hazardous waste management support services at the INEL; his duties included preparing task order proposals, negotiating contracts, performing cost and schedule accountability/reporting, and maintaining responsiveness to client needs.

U.S. Department of Energy - Albuquerque Operations Office
Albuquerque, New Mexico
1982–1984

Within the Facility Design and Safety Analysis Branch of the ES&H Division of DOE-AL, as project engineer, Mr. Fredrickson developed criteria, reviewed designs, performed safety analyses, assessed risks, appraised preoperational safety, and evaluated NEPA compliance for nonreactor facilities (e.g., plutonium processing, tritium, high explosives, and waste management) in the DOE complex. He conducted

radiological safety analyses for over 50 such facilities and received the DOE "Monetary Award for Special Act or Service" for his accomplishments in these areas.

Westinghouse Electric Corporation
WIPP Project
Albuquerque, New Mexico
1979–1982

As a senior engineer, Mr. Fredrickson managed the Safety Analysis Report Program for the WIPP project; developed radiological assessment portions of the final environmental impact statement; participated in developing waste acceptance and packaging criteria; and conducted operational, postulated accident, criticality safety, and transportation risk analyses. He directed the assessment of potential long-term hydrogeologic transport of radionuclides to the biosphere following closure. He represented the project in these areas to the DOE, state of New Mexico, and the National Academy of Sciences.

Westinghouse Electric Corporation
Nuclear Technology Division
Monroeville, Pennsylvania
1973–1979

As safety engineer, Mr. Fredrickson coordinated the resolution of a wide range of generic safety and licensing issues for the Westinghouse nuclear steam supply system, including the characterization and evaluation of turbine- and tornado-generated missiles, mechanical system safety, post-accident environmental qualification, and water hammer. He prepared portions of 36 SARs for Westinghouse reactors and was a consultant to the International Atomic Energy Agency and American National Standards Institute committee chairman on reactor design for natural phenomena-generated missiles.

In a volunteer capacity while with Westinghouse, Mr. Fredrickson participated in an outreach program directed at educating the general public on issues associated with nuclear power. Associated activities included television, radio and print media events, debates on college campuses, and briefings with policy-makers in more than 30 states over a six-year period.