Homework 1

Problem 1

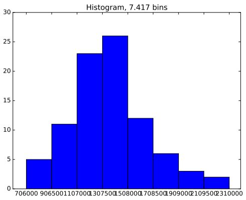

Figure 1: Sturges

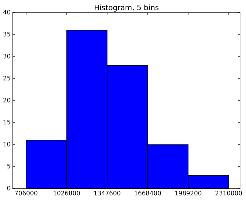

Figure 2: 5 bins

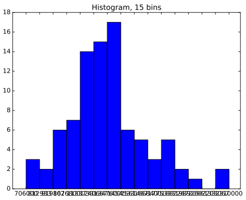

Figure 3: 15 bins

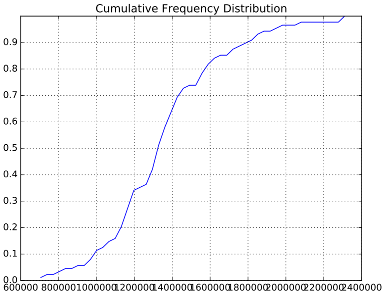

Figure 4: CDF

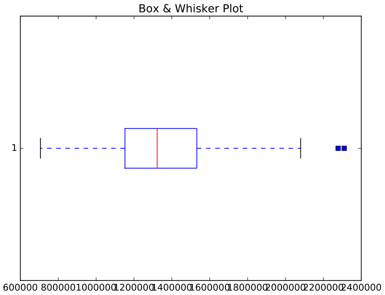

Figure 5: Boxplot

The boxplot (figure 5) contains two outliers: 2,278,000 and 2,310,000, both above the 75th-quartile plus the maximum whisker length of $q75 + 1.5 * iqr$. The lower whisker only extends to 706,000 because this is within the maximum whisker length.

Here are the descriptive statistics (question 1c):

```
mean, 1352204.54545
median, 1322500.0
var, 98171291013.6
q25, 1152250.0
q50, 1322500.0
q75, 1531750.0
iqr, 379500.0
skew, 0.661353249864
qskew, 0.102766798419
```

Problem 2

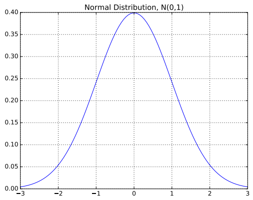

Figure 6: Normal

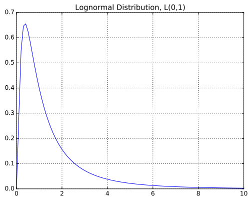

Figure 7: Lognormal

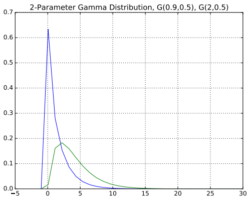

Figure 8: Gamma

Below is the python code used to create the plots.

Listing 1: statswrap.py statistics wrapper

```python
from __future__ import division
import matplotlib.pyplot as plt
import numpy
import pandas
import os
import statsmodels.api as sm
import math
from scipy.stats import norm, lognorm, gamma

class Statistics:

  def __init__(self, file, col, output):
    self.filename = os.path.splitext(file)[0]
    self.file = pandas.read_csv(file)
    self.values = self.file[col]
    self.output = output

  def get_stats(self):
    """ mean, median, var, iqr, skew, qskew, quartiles = Statistics.get_stats()"""
    """ where 'quartiles' will provide a list [q25, q50, q75] """
    mean = self.values.mean() # [0]
    median = self.values.median() # [0]
    var = self.values.var() # [0]
    q25 = self.values.quantile(q=0.25) # [0]
    q50 = self.values.quantile() # [0]
    q75 = self.values.quantile(q=0.75) # [0]
    iqr = q75-q25
    qskew = ((q75-q50)-(q50-q25))/iqr
    skew = self.values.skew() # [0]
    return (mean, median, var, iqr, skew, qskew, [q25, q50, q75])

  def write_stats(self):
    mean, median, var, iqr, skew, qskew, quartiles = self.get_stats()
    with open(self.output + 'stats.txt', 'w') as f:
      f.write('mean, ' + str(mean) + '\n')
      f.write('median, ' + str(median) + '\n')
      f.write('var, ' + str(var) + '\n')
      f.write('q25, ' + str(quartiles[0]) + '\n')
      f.write('q50, ' + str(quartiles[1]) + '\n')
      f.write('q75, ' + str(quartiles[2]) + '\n')
      f.write('iqr, ' + str(iqr) + '\n')
      f.write('skew, ' + str(skew) + '\n')
      f.write('qskew, ' + str(qskew))

  def get_sturges(self):
    n = len(self.file.index)
    self.nbins = 1+3.3*math.log10(n)
    smallest = self.values.min() # [0]
    largest = self.values.max() # [0]
    binwidth = (largest-smallest)/self.nbins
    x = smallest.copy()
    bins = []
    while x < largest:
      bins.append(x)
      x+=binwidth
    bins.append(x)
    return bins

  def plot_hist(self, bins):
    q = self.values.values
    figfile = ''
    if bins == 'sturges':
      figfile = (self.output + 'histogram_sturges.pdf')
    if bins != 'sturges':
      self.nbins = bins
      figfile = (self.output + 'histogram_' + '{0}' + '.pdf').format(self.nbins)
    fig, ax = plt.subplots()
    counts, bins, patches = ax.hist(q, bins=bins)
    ax.set_title(('Histogram, {0:.4g} bins').format(self.nbins))
    ax.set_xticks(bins)
    fig.savefig(figfile)
    # plt.show()

  def cumul_freq_dist(self, filename='cumul_freq_dist.pdf'):
    n = len(self.file.index)
    sample = self.values.values.ravel()
    ecdf = sm.distributions.ECDF(sample)
    x = numpy.linspace(self.values.min(), self.values.max()) #[0]
    y = ecdf(x)
    fig, ax = plt.subplots()
    ax.plot(x,y)
    ax.set_title('Cumulative Frequency Distribution')
    ax.set_yticks(numpy.arange(0,1,0.1))
    plt.grid()
    fig.savefig(self.output + filename)
    # plt.show()

  def boxplot(self, filename='boxplot.pdf'):
    data = self.values.values.ravel()
    fig, ax = plt.subplots()
    ax.boxplot(data, 0, 'rs', 0)
    ax.set_title('Box & Whisker Plot')
    fig.savefig(self.output + filename)
    # plt.show()


  def norm_dist(self, mean=0, var=1, xstart=-3, xstop=3, xnum=100, filename='norm_dist.pdf'):
    x_axis = numpy.linspace(xstart, xstop, xnum) # xnum steps
    fig, ax = plt.subplots()
    ax.plot(x_axis, norm.pdf(x_axis,mean,var))
    plt.grid()
    ax.set_title('Normal Distribution, N({0},{1})'.format(mean,var))
    fig.savefig(self.output + filename)
    # plt.show()

  def lognorm_dist(self, mean=0, var=1, xstart=0, xstop=10, xnum=100, filename='lognorm_dist.pdf'):
    x_axis = numpy.linspace(xstart, xstop, xnum) # xnum steps
    fig, ax = plt.subplots()
    ax.plot(x_axis, lognorm.pdf(x_axis,var,0,numpy.exp(mean)))
    plt.grid()
    ax.set_title('Lognormal Distribution, L({0},{1})'.format(mean,var))
    fig.savefig(self.output + filename)
    # plt.show()

  def gamma_dist(self, klist=[0.9,2], lam=0.5, xstart=-1, xstop=10, xnum=30, filename='gamma_dist.pdf'):
    x_axis = numpy.linspace(xstart, xstop, xnum) # xnum steps
    try:
      scale = 1/lam
    except ZeroDivisionError:
      scale = 1/0.5
      print 'ZeroDivisionError: gamma_dist() lambda value set
    to 0.5 as default'
    fig, ax = plt.subplots()
    for k in klist:
      ax.plot(x_axis, gamma.pdf(x_axis,k,scale=scale))
    plt.grid()
    ax.set_title('2-Parameter Gamma Distribution, G({0},{1}), G({2},{3})'.format(klist[0],lam,klist[1],lam))
    fig.savefig(self.output + filename)
    # plt.show()

if __name__ == "__main__":

  stats = Statistics('hw1given.csv', 'peak_streamflow_cfs', 'probs_1_2/')

  # Problem 1a
  sturges = stats.get_sturges()
  stats.plot_hist('sturges')
  stats.plot_hist(5)
  stats.plot_hist(15)

  # Problem 1b
  stats.cumul_freq_dist()

  # Problem 1c
  stats.write_stats()

  # Problem 1d
  stats.boxplot()

  # Problem 2
  stats.norm_dist(0,1,-3,3,100)
  stats.lognorm_dist(0,1,0,10,100)
  stats.gamma_dist([0.9,2],0.5,-1,30,30)
```