Class Example 1. Single Sample Descriptive Statistics

(a) Summary Statistics and Box-Plots

You are working in factory producing hand held bicycle pumps and obtain a sample of 174 bicycle pump weights in grams, as in the data file *Pumps.csv*. Carry out descriptive statistics as follows:

(i) Load the data file.

```
using DataFrames
data = readtable("Pumps.csv")
```

(ii) Output the sample size, sample mean and sample standard deviation.

```
(length(data[1]),mean(data[1]),std(data[1]))
```

(iii) View basic summary statistics.

```
using StatsBase
summarystats(data[1])
```

(iv) Create a box-plot of the data and comment on the structure of the data.

```
using PyPlot
PyPlot.boxplot(data[1])
```

(v) Present the summary statistics and the box-plot of the data in a neatly formatted Julia notebook describing the data set.

Summary statistics and a box-plot of 174 pump weights

In [19]:
```
using DataFrames, StatsBase, PyPlot
data = readtable("Pumps.csv");
```

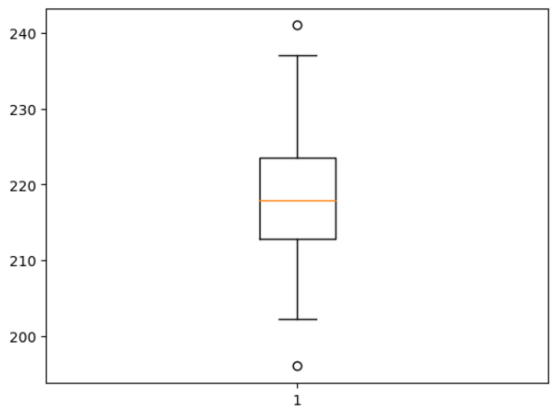
In [22]:
```
(length(data[1]),mean(data[1]),std(data[1]))
```

Out[22]:
```
(174,218.11448275862068,7.71757011365874)
```

In [23]:
```
summarystats(data[1])
```

Out[23]:
```
Summary Stats:
Mean:         218.114483
Minimum:      196.020000
1st Quartile: 212.825000
Median:       217.945000
3rd Quartile: 223.542500
Maximum:      241.020000
```

In [24]:
```
PyPlot.boxplot(data[1]);
```



A brief summary of the data:

As can be seen the mean pump weight is at 218 grams and the standard deviation is at 7.7 grams. The data appears to be symmetric with only two noted outliers out of 174 observations.

(b) Empirical Cumulative Distribution Function

(i) Key in the code below and run it.

```
using Distributions, StatsBase,PyPlot
zRV = Normal(20,5)
numSamples = 15
data = rand(zRV,numSamples)
estimatedCDF = ecdf(data)

grid = 0:0.1:40
PyPlot.plot(grid,estimatedCDF(grid))
PyPlot.plot(grid,cdf(zRV,grid))
```

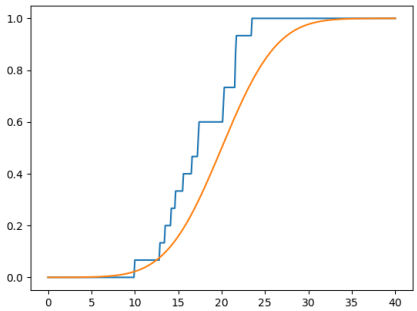


Figure 1: The resulting CDF of a theoretical Normal distribution with $\mu = 20$ and $\sigma = 5$ plotted against an ECDF of randomly generated samples from the same distribution.

(ii) Now modify the number of samples seeing how the ECDF converges to the true CDF as the number of samples increases.

(iii) Change the distribution (zRV) to a distribution of your choice, e.g. Exponential or Uniform. Now run again to view the resulting graphs. Note that you may need to change the values of grid, appropriately.

(c) Histograms and Kernel Density Estimation.

The code below generates random variables from a *mixture of two normal distributions* with pdf represented by the blue curve below (left). It then plots a kernel density estimate of the data (red) and a histogram of the data.

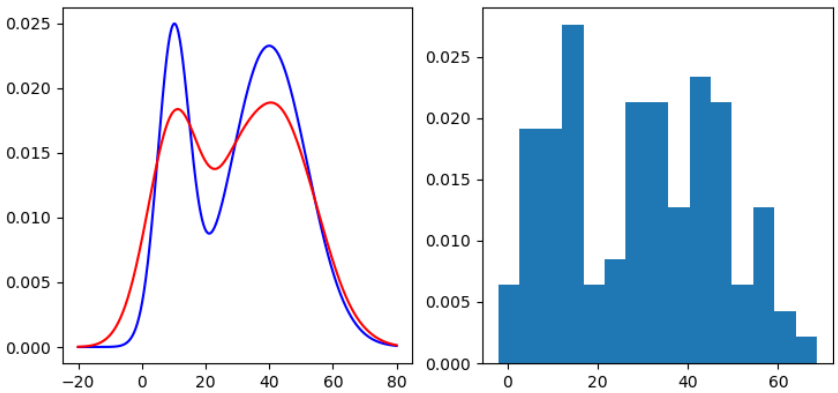


Figure 2: On left: Blue is theoretical pdf and red is a kernel density estimate (depends on data). On Right: A simple histogram.

Key in this code and experiment with increasing the number of samples. You may also try changing other parameters such as the means, variances and p.

```
using Distributions, KernelDensity, PyPlot

mu1 = 10;  sigma1=5
mu2 = 40;  sigma2=12

z1 = Normal(mu1,sigma1)
z2 = Normal(mu2,sigma2)

p = 0.3

#mixRV is a mixture of the Normal z1 and the Normal z2.
function mixRv()
    ind = (rand() <= p)
    if ind
        return rand(z1)
    else
        return rand(z2)
    end
end

#The pdf of a mixture is a mixture of the pdfs
function actualPDF(x)
    return p*pdf(z1,x) + (1-p)*pdf(z2,x)
end

numSamples = 100
data = [mixRv() for _ in 1:numSamples]

grid = linspace(-20,80,1000)

#Plot the actual pdf (in blue) against the estimated KDE (in red)
subplot(121)
pdfActual = actualPDF.(grid)
PyPlot.plot(grid,pdfActual,"blue")

kdeDist=kde(data)
pdfKDE=pdf(kdeDist,grid)
PyPlot.plot(grid,pdfKDE, "red")

#for comparison, plot a histogram
subplot(122)
PyPlot.plt[:hist](data,15,normed="True")
```

Class Example 2. The Correlation Coefficient and Bivariate Normal Distribution

Consider a dataset consisting of the daily maximum temperature (in °C) recorded at Brisbane and Gold Coast each day from the start of 2015 to the middle of Febuary 2017. The data is from the Australian Bureau of Meteorology, http://www.bom.gov.au/.
Denote the observations for Brisbane and Gold Coast (respectively) by:

$$x_1, \ldots, x_n \qquad \text{and} \qquad y_1, \ldots, y_n.$$

Here $n = 777$. We will now perform the following:

(a) Load the data file and ensure you have the right dimensions.

(b) Calculate the following from the data:

(i) The sample means for temperatures at both locations.

(ii) The sample standard deviations for temperatures at both locations.

(iii) The correlation coefficient estimate.

(c) Use these estimates to describe the nature of the data in about 3 lines.

(d) Draw a scatter plot of the data.

(e) Assume the data comes from a bivariate Normal distribution.

(i) Draw a contour plot diagram of the estimated distribution.

(ii) Draw a 3D graph of the estimated distribution.

(iii) Write an expression for the probability of having a day in Brisbane with temperature less than 30 degrees and temperature in Gold Coast greater than 25 degrees.

Solution:

(a) Load the .csv file as follows:

```
data = readcsv("BrisGCtemp.csv")
```

Then noticing that the data points begin on the second row and the respective temperatures are in the 4'th and 5'th columns, we extract arrays for Brisbane and Gold Coast as follows:

```
Bris = convert(Array{Float64,1},data[2:end,4])
GC = convert(Array{Float64,1},data[2:end,5])
length(Bris), length(GC)
```

(b) (i) The formulas for calculating the sample mean are as follows:

$$\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n} \qquad \text{and} \qquad \overline{y} = \frac{\sum_{i=1}^{n} y_i}{n}$$

```
mean(Bris),mean(GC)
```

(ii) The **sample variance** is calculated using:

$$s^2 = \frac{\sum_{i=1}^{n} (x_i - \overline{x})^2}{n-1} = \frac{\sum_{i=1}^{n} x_i^2 - n\,\overline{x}^2}{n-1}$$

and the **sample standard deviation, s**, is the positive square root of the sample variance. We calculate this here using both the formula and the built-in Julia std() function.

```
sqrt(
        (sum([x^2 for x in Bris]) - length(Bris) * mean(Bris)^2)
        /(length(Bris)-1)),
std(Bris),
sqrt(sum([(y-mean(GC))^2 for y in GC])  / (length(GC)-1)),
std(GC)
```

(iii) The correlation coefficient estimate is calculated using:

$$r_{xy} = \frac{\sum_{i=1}^{n}(y_i - \overline{y})(x_i - \overline{x})}{\left[\sum_{i=1}^{n}(y_i - \overline{y})^2 \sum_{i=1}^{n}(x_i - \overline{x})^2\right]^{1/2}}$$

In Julia we use the following:

```
cor(Bris,GC)
```

(c) We see that the estimated mean temperature in Brisbane is 27.15 and the estimated mean temperature in the Gold Coast is 26.16. These are with corresponding sample standard deviations 4.02 and 3.52. The correlation coefficient is 0.92 hinting that on days where the temperature is high in Brisbane, it is more likely to be high in the Gold Coast, and vice versa.

Note: Since the data is daily data over a long period, it is very plausible that it encompasses seasonal effects. Hence when considering the variation of the data (standard deviations of around 4) we need to be cognisant of the fact that much of the variation is perhaps due to seasonal effects. Separating the seasonal effects and random effects, is a matter of further study.

(d) We can visualise the data using a scatter plot:

```
using PyPlot
PyPlot.plot(Bris,GC,".", markersize=2)
xlabel("Brisbane Daily Max Temp (C)")
ylabel("Gold Coast Daily Max Temp (C)");
```

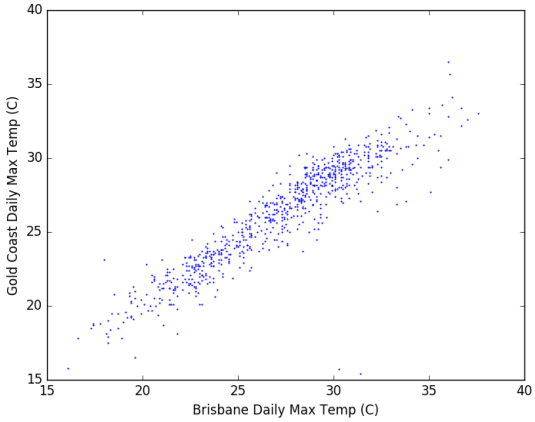
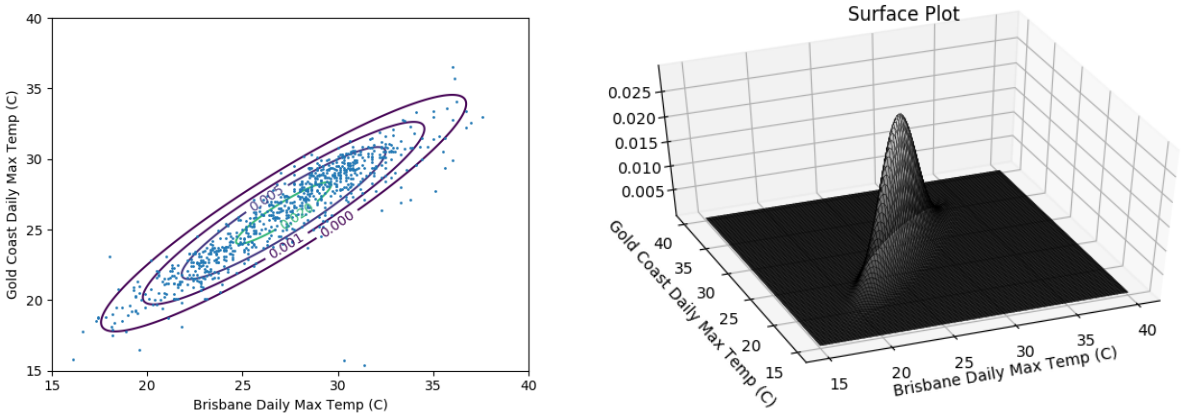


Figure 3: Brisbane Max Temp vs Gold Coast Max Temp

(e) Understanding the code below is beyond the scope of the course. Nevertheless, it is good to see as it provides a neat representation of the data together with the fitted bivariate Normal Distribution, answering (i)-(ii).

```
using PyPlot

function biVariateGauss(x, mu, sigma)
    return ((1 / (2pi)) * det(inv(sigma))^(1/2)) * exp(- (x - mu)'
            * inv(sigma) * (x - mu))[1]
end

n = 200
mu = [mean(Bris); mean(GC)]
sigma =  [std(Bris)^2 cor(Bris,GC)*std(GC)*std(Bris);
          cor(Bris,GC)*std(GC)*std(Bris) std(GC)^2]

x = linspace(15, 40, n)
y = linspace(15, 40, n)
xgrid = repmat(x', n, 1 )
ygrid = repmat(y, 1, n )
grid = [xgrid[:] ygrid[:]]'
z = [biVariateGauss(grid[:,i], mu, sigma) for i in 1:size(grid, 2)]
zgrid = reshape(z, n, n)

PyPlot.figure("Contour plot")
xlim(15,40)
ylim(15,40)
CS = PyPlot.contour(xgrid, ygrid, zgrid, levels=[0.0001,0.001,
                                                 0.005, 0.02, 0.03])
PyPlot.clabel(CS, inline=1, fontsize=10)
PyPlot.plot(Bris, GC, ".", markersize=2, label="scatter")
xlabel("Brisbane Daily Max Temp (C)")
ylabel("Gold Coast Daily Max Temp (C)")
title("Contour Plot")

z = zeros(n,n)
for i in 1:n
    for j in 1:n
        z[i,j] = biVariateGauss([x[i],y[j]], mu, sigma)
    end
end

fig = figure("pyplot_surfaceplot",figsize=(5,10))
ax = fig[:add_subplot](2,1,1, projection = "3d")
ax[:plot_surface](xgrid, ygrid, z, rstride=2,edgecolors="k",
                  cstride=2, cmap=ColorMap("gray"), alpha=0.8,
                  linewidth=0.25)
xlabel("Brisbane Daily Max Temp (C)")
ylabel("Gold Coast Daily Max Temp (C)")
title("Surface Plot")
```



(a) Contour plot (b) 3D surface plot

Figure 4: Brisbane Max Temp vs Gold Coast Max Temp

(iii) Write an expression for the probability of having a day in Brisbane with temperature less than 30 degrees and temperature in Gold Coast greater than 25 degrees.

$$\mathbb{P}\Big(X < 30, Y > 25\Big) = \int_{x=-\infty}^{30} \int_{y=25}^{\infty} f_{X,Y}(x, y; s_x, s_y, \overline{x}, \overline{y}, \hat{\rho})\, dx\, dy,$$

where $f_{X,Y}(\cdot)$ is the bivariate normal density:

$$f_{XY}(x, y; \sigma_X, \sigma_Y, \mu_X, \mu_Y, \rho) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}} \times \exp\left\{\frac{-1}{2(1-\rho^2)}\left[\frac{(x-\mu_X)^2}{\sigma_X^2} - \frac{2\rho(x-\mu_X)(y-\mu_Y)}{\sigma_X\sigma_Y} + \frac{(y-\mu_Y)^2}{\sigma_Y^2}\right]\right\}.$$

Class Example 3. Statistical Inference Ideas.

A farmer wanted to test whether a new fertilizer was effective at increasing the yield of her tomato plants. She took 20 plants, kept 10 as controls and treated the remaining 10 with the new fertilizer. After two months, she harvested the plants and recorded the yield of each plant (in kg), as shown in the following table:

Control	4.17	5.58	5.18	6.11	4.5	4.61	5.17	4.53	5.33	5.14
Fertilizer	6.31	5.12	5.54	5.5	5.37	5.29	4.92	6.15	5.8	5.26

From this data, the group of plants treated with the fertilizer had an average yield of 0.494 kg greater than the control group. One could argue that this difference is due to the effects of fertilizer. We will now investigate if this is a reasonable assumption.

Let us assume for a moment that the fertilizer has no effect on plant yield (it is a placebo/has no added nutrients). In such a scenario, we actually have 20 observations from the **same group** (non-nutrient enriched plants), and the average difference is purely the result of random chance.

We can investigate this, by taking all possible combinations of 10 samples from our group of 20 observations, and counting how many of these combinations results in a sample mean greater than the mean of our treatment group. Dividing this total by the total number of possible combinations, we can obtain a proportion, and hence likelihood, that the difference observed was due purely to random chance.

First calculate the number of ways of sampling $r = 10$ unique items from a total of $n = 20$. This is given by,

$$\binom{n}{r} = \frac{n!}{r!(n-r)!} = \binom{20}{10} = 184,756.$$

This number of possible samples is computationally managable, and hence we'll use Julia's combinatorics package to enumerate all 184,756 combinations.

```
using Combinatorics, DataFrames
# Import data
data = readtable("Fertilizer.csv")
control = data[1]
fertilizer = data[2]

# Concantenate the data, and collect all 20C10 combinations.
x = collect(combinations([control;fertilizer],10))

# Let's just make sure that the number of combinations is correct
println("Number of combinations: ",length(x))

# Construct a vector of means for each combination, and calculate the
# proportion of times these means are >= than the mean of treated group.
pvalue = sum([mean(_) >= mean(fertilizer) for _ in x])/length(x)
```

We can see that from this data, only 2.39% of all possible combinations have a mean greater or equal to our treated group. Therefore there is significant statistical evidence that the fertilizer increases the yield.

Question 1. Joint Probability Mass Function

Consider the function $p_{XY}(\cdot,\cdot)$:

x	y	$p_{XY}(x,y)$
1.0	1.0	1/4
1.5	2.0	1/8
1.5	3.0	1/4
2.5	4.0	1/4
3.0	5.0	1/8

Determine the following:

(a) Show that $p_{X,Y}$ is a valid probability mass function.

(b) $P(X < 2.5,\ Y < 3)$.

(c) $P(X < 2.5)$.

(d) $P(Y < 3)$.

(e) $P(X > 1.8,\ Y > 4.7)$.

(f) $E(X),\ E(Y),\ V(X),\ V(Y)$.

(g) Are X and Y independent random variables?

(h) $P(X + Y \leq 4)$.

Question 2. More Fun With Two Random Variables

Let X and Y be independent random variables with $E(X) = 2, V(X) = 5, E(Y) = 6,\ V(Y) = 8$. Determine the following:

(a) $E(3X + 2Y)$.

(b) $V(3X + 2Y)$.

Assume now further to the above that X and Y are normally distributed and determine the following:

(c) $P(3X + 2Y < 18)$.

(d) $P(3X + 2Y < 28)$.

(e) Verify (c) and (d) using Julia code, where for each case you generate a million X's and a million $Y's$ and simulate the linear combination $3X + 2Y$.

(f) Assume now that the random variables come from another distribution (not Normal), but keep the same means and variances. Are your answers for (c) and (d) likely to change? How about your answers for (a) and (b)?

(g) Assume now that X and Y are Normally distributed but are not independent, but rather $Cov(X, Y) = 5$. Write an explicit expression using a double integral for $P(X < 2, Y > 7)$.

Question 3. Rise of the machines

A semiconductor manufacturer produces devices used as central processing units in personal computers. The speed of the devices (in megahertz) is important because it determines the price that the manufacturer can charge for the devices. The file (*6-42.csv*) contains measurements on 120 devices. Construct the following plots for this data and comment on any important features that you notice.

(a) Histogram.

(b) Box-plot.

(c) Kernel Density Estimate.

(d) Empirical cumulative distribution function.

Further, compute:

(e) The sample mean, the sample standard deviation and the sample median.

(f) What percentage of the devices has a speed exceeding 700 megahertz?

Question 4. The thickest rod

Eight measurements were made on the inside diameter of forged piston rings used in an automobile engine. The data (in millimetres) is:

74.001, 74.003, 74.015, 74.000, 74.005, 74.002, 74.005, 74.004.

Use the Julia function below to construct a normal probability plot of the piston ring diameter data. Does it seem reasonable to assume that piston ring diameter is normally distributed?

```
using PyPlot, Distributions, StatsBase

function NormalProbabilityPlot(data)
    mu = mean(data)
    sig = std(data)
    n = length(data)
    p = [(i-0.5)/n for i in 1:n]
    x = quantile(Normal(),p)
    y = sort([(i-mu)/sig for i in data])
    PyPlot.scatter(x,y)
    xRange = maximum(x) - minimum(x)
    PyPlot.plot([minimum(x)- xRange/8,maximum(x) + xRange/8],
                [minimum(x)- xRange/8,maximum(x)+ xRange/8],
                                color="red",linewidth=0.5)
    xlabel("Theoretical quantiles")
    ylabel("Quantiles of data");
    return
end
```

Question 5. A non-flat earth

In 1789, Henry Cavendish estimated the density of the Earth by using a torsion balance. His 29 measurements are in the the file (*6-122.csv*), expressed as a multiple of the density of water.

(a) Calculate the sample mean, sample standard deviation, and median of the Cavendish density data.

(b) Construct a normal probability plot of the data. Comment on the plot. Does there seem to be a "low" outlier in the data?

(c) Would the sample median be a better estimate of the density of the earth than the sample mean? Why?

Question 6. Normal Confidence Interval

A normal population has a mean 100 and variance 25. How large must the random sample be if you want the standard error of the sample average to be 1.5?

Question 7. Fill in the blanks

A random sample has been taken from a normal distribution. Output from a software package follows:

Variable	N	Mean	SE Mean	StDev	Variance	Sum
x	?	?	1.58	6.11	?	751.40

(a) Fill in the missing quantities.

(b) Find a 95% CI on the population mean under the assumption that the standard deviation is known.

Question 8. More on the randomization test

Reproduce class example 3. Now modify the data so that the yield of the Fertilizer is decreased by exactly 0.5 kg per observation (i.e. the first observation is 5.81, the second is 4.62 and so fourth). What are the results now? How do you interpret them?