In [1]:
```
import pandas as pd
import numpy as np
import math as mt
import matplotlib.pyplot as plt
import scipy.stats as st
from scipy.stats import norm
from scipy.stats import pearsonr
import statsmodels.formula.api as smf
%matplotlib inline
```

Exam GEO4300, 23.11.2020

(1) Random variable parameter estimation

(a) The expected value of X

For a discrete random variable and the corresponding probabilities $f(x_j)$:

$E(X) = \sum x_j f(x_j)$

In [37]:
```
# A) The expected value
E_x = (-1*1/3) + (3*1/2) + (4*1/6)
print('The expected value E(x): %.2f ' % E_x)
```

```
The expected value E(x): 1.83
```

(b) The variance

For a discrete variable: $\sigma_s^2 = \sum \frac{(x-\mu)^2}{n}$

The variance of the random variable X is defined as: $V(x) = E[x - E(x)]^2 = E(x - \mu)^2$ Used here.

In [53]:
```
# B) The variance

#Can use both formulas.
#var = (((-1-E_x)**2)*(1/3) + ((3-E_x)**2)*(1/2) + ((4-E_x)**2)*(1/
6))

V_x = (1/3)*(-1-E_x)**2 + (1/2)*(3-E_x)**2 + (1/6)*(4-E_x)**2
print('The variance V(x):  %.2f ' % V_x)
```

```
The variance V(x):  4.14
```

(c) The mode

For a discrete variable: The mode is the x value associated with $Max_{i=1}^{n} f(x)$

In [32]:
```
# C) The mode. The most frequently occuring value in a set of discr
ete data

p_max = 1/2    #The highest probability
x_mode = 3;

print('The mode: %.2f' % x_mode)
```

```
The mode: 3.00
```

(d) The coefficient of variation

The coefficient of variation: $Cv = \frac{V(x)}{E(x)}$

In [36]:
```
# D) The coefficient of variation

Cv = np.sqrt(V_x)/E_x

print('The coefficient of variation Cv:  %.2f ' % Cv)
```

```
The coefficient of variation Cv:  1.11
```

(2) Frequency analysis and linear regression

(a) The probability to observe at leat one 100-year flood or larger whitin a period of 10 years:

Probability one flood at least one time in 10 years: $P = 1 - (1 - \frac{1}{T})^n$

In [51]:
```
T = 100
n = 10

p = 1/T

P_one_flood_10_years = 1 - (1-(1/T))**n


print('Probability of at leat one flood in 10 years, with T = 100 y
ears:  %.3f \n' % P_one_flood_10_years)
```

```
Probability of at leat one flood in 10 years, with T = 100 years:
0.096
```

(b)

The assumption of a simple linear regression that is violated in this analysis: The multivariate normality/that the data is normally distributed. This assumption can be evaluated with a QQ-plot. The standarized residuals and theoretical quantiles should fall on a line, and we see that the residulals falls on the line for approximately x and y = -1 to x and y = 2, but that the residuals for the larger floods., especially larger than 100 m3/s does not fall on the line, which means that they are not normally distributed. We are usually interested in the larger floods, so this is problematic.

Strategies to improve the analysis: Get larger sample size/more data, but this is difficult. We should try other methods using different distributions (not normal distribution), such as Log-Normal, Extreme-value type I (Gumble), Pearson III or Log-Pearson III. Plot of the streamflow vs the return period can show how well fitted the model is, in addition to the QQ-plots. Then, expert knowledge is needed in order to choose the most suitable model. Should the model predicting the largest runoff be used? This may cause uneccessary high mitigation and protection costs, so it may be more effective to use a model in between the most and least conservative.

(3) Confidence intervals

n = 30, mean = 145, variance = 20

In [47]:
```
# (A, i) The true variance is unknown, estimated as 30. Confidence
interval for the mean.

var = 20                                    # Estimated variance
mean =145;                                  # Estimated mean
alpha = 0.05                                # Significance level
n = 30                                      # Number of observations
std_mean = np.sqrt(var)/mt.sqrt(n)          # Standard deviation around
mean

t = st.t.ppf(1-alpha/2, df = n-1);          # Critical value t, use t-ta
ble because of unknown variance

l = mean - (t *std_mean)                     # Lower limit
u = mean + (t *std_mean)                     # Upper limit

print("The 95%% confidence interval for the mean is [%.1f, %.1f]."%
(l,u))
```

```
The 95% confidence interval for the mean is [143.3, 146.7].
```

In [49]:
```
# (A,ii) True variance is known. Confidence interval for the mean.

mean = 145
var = 20
alpha = 0.05
n = 30
std_mean = np.sqrt(var)/mt.sqrt(n)    # Standard deviation around me
an

z = st.norm.ppf(1-alpha/2);           # Critical value z, use z-tabl
e because the variance is known.


l = mean - (z *std_mean)              # Lower limit
u = mean + (z *std_mean)              # Upper limit

print("The 95%% confidence interval for the mean is [%.1f, %.1f]."%
(l,u))
```

```
The 95% confidence interval for the mean is [143.4, 146.6].
```

(b) The reason for the difference of results in part (i) and part (ii):

The results in (i) and (ii) are different because we use different distributions, z and t. The z-distribution is steeper and has more mass in the middle, which gives larger confidence interval. The t is flatter and gives smaller confidence interval. When the number of samples increases, the t distribution approaches the z distribution, because the degree of freedom increases. Thats why we can use the z-distribution for large sample sizes or when true variance is known.

In [50]:
```
# (C) The 95% confidence interval on the variance.

chi_1 = st.chi2.ppf(alpha/2, df = n-1)           # Use the Chi-Squa
re table because we test for the variance
chi_2 = st.chi2.ppf((1-alpha/2), df = n-1)

l = ((n-1)*(var))/chi_2
u = ((n-1)*(var))/chi_1

print("The 95%% confidence interval for the variance is [%.1f, %.1f
]." %(l,u))
```

```
The 95% confidence interval for the variance is [12.7, 36.1].
```

(4) Machine learning

(a)

It is common to split the dataset into a training set and a test set when doing supervised machine learning. You can for example use 2/3 of the data as the training set and 1/3 as the test set. This is because we want to train the model on a set of data to learn it to predict the output which is part of the data used to construct a learner. After training the model, we want to test if the model is able to predict the output using a different dataset. We use the test set to evaluate the model. The model is a result of the data we put in, but we want to predict other observations. After the learning of the model, we can evaluate the model by the training error (the prediction error of the training set), but this error underestimates the test error. After using the test training, the model can be evaluated from the test-error (the prediction error of the test set).

b)

It is important to control the complexity of the model so that it is not underfitting (too simple, large test and train error) or overfitting (too complex,large test error, small train error). Need to find the sweet spot: The model complexity that gives the smallest test error.

(5) Fourier transformation

(a)

You could use the run test method to test if there is a significant trend in X_t.

Step 1: Calculate the mean or median of X

Step 2: Compare each value of X to the mean.

If X_i > X_mean, give a sign '+'. If X_i < X_mean, give a sign '-'.

Step 3: Calculate the number of values with '+' and '-', denote them as n1 and n2.

Step 4: Calculate the number of positve and negative runs, denote is as R.

Step 5: For large n1,n2 > 10: The distribution of R is approximated by a normal distribution with a mean of:

$$\mu = \frac{2*n1*n2}{n1+n2} + 1$$

and variance of:

$$\sigma^2 = \frac{2*n1*n2*(2*n1*n2-n1-n2)}{(n1+n2)^2*(n1+n2-1)}$$

Step 6: Use μ and σ^2 to calculate the test statistics (z-test)

$$Z = \frac{R-\mu}{\sigma}$$

If $|Z| > Z_{1-a/2}$, we have a significant trend at significance level a.

You can also do the Mann-Kendall test method to test if the trend is significant, or do a hypothesis test on the significance of the slope of the ordinary least square regression line to test if there is a significant trend.

(b)

The existence of a periodic component in the data may be investigated by a Fourier analysis. Changes the signal from time domain to frequency domain.

Figure A shows the Fourier transform of X_t.

In [ ]: