EXAM

GEO4300 – Geophysical Data Science

Candidate number:
15403

University of Oslo

Autumn 2020

1. Random Variable Parameter Estimation

```
x = [-1, 3, 4]

#Expected value:
EV1 = x[0]*(1/3)
EV2 = x[1]* (1/2)
EV3 = x[2]* (1/6)
EV = EV1 + EV2 + EV3
print ("Expected value: %.3f" %(EV))

#Variance:
V1 = (x[0]**2)*(1/3)
V2 = (x[1]**2)*(1/2)
V3 = (x[2]**2)*(1/6)
V4 = V1 + V2 + V3
V = V4 - (EV**2)
print ("Variance: %.3f" %(V))

#Mode:
print ("Mode: 3 due to probability of 1/2 (reacurrs most)")

#Std: #used this to find CV
std = math.sqrt(V)
#print ("Std:", (std))

#Mean: #used this to find CV
mean = statistics.mean(x)
#print ("Mean:", (mean))

#Coefficient of variation:
CV = std/mean
print ("Coefficient of Variation: %.3f" %(CV))
```

```
Expected value: 1.833
Variance: 4.139
Mode: 3 due to probability of 1/2 (reacurrs most)
Coefficient of Variation: 1.017
```

2. Frequency analysis

a.)

```
#100 year flood:
flood_100 = 1-(1/100)

#Probability of no flood:
no_flood = flood_100**10

#Probability of at least one flood:
flood = 1-no_flood

print("The probability to observe at least one 100-years flood or larger within a period of 10 years is: %.4f"
```

```
The probability to observe at least one 100-years flood or larger within a period of 10 years is: 0.0956
```

b.) A QQ-plot can be used to evaluate the normality assumption of the residuals. This is done by comparing the residuals to "ideal" normal observations. Due to the shape (note: x and y - axis) of the observations in the QQ-plot I can assume that the assumption of a normal distribution is violated.

The simple linear regression model is representing the data well, but not the QQ-plot. The QQ-plot is displaying a poor fit to the data. This can be corrected by using another model. There are several extreme events/outliers which suggest the use of a model which takes more extreme events into account. To improve the results its best to try different models and see which model that fits the majority of the data best. Some examples are the normal distribution, lognormal distribution, Gumbel distribution, Pearson type III distribution and log-Pearson type III.

3. Confidence Intervals

a.)
(i)

```
#3. Confidence Intervals:
#a.) A normal distribution is assumed

n = 30 #sample size
m = 145 #sample mean
v = 20 #variance
a = 0.05 #significance level

#(i) - variance unknown: t-test

stdm = np.sqrt(v/n)

t = st.t.ppf(1-a/2, n-1) #t-critical

l = m - t*stdm
u = m + t*stdm
print("The lower confidence interval is %.2f and the upper confidence interval is %.2f" %(l,u))
```

```
The lower confidence interval is 143.33 and the upper confidence interval is 146.67
```

(ii)

```
#(ii) - variance known: z-test:

z = st.norm.ppf(1-a/2) #z-critical

l = m-z*stdm #lower interval
u = m+z*stdm #upper interval

print("The lower confidence interval is %.2f and the upper confidence interval is %.2f" %(l,u))
```

```
The lower confidence interval is 143.40 and the upper confidence interval is 146.60
```

b.)
When the variance is known and not estimated, we are more certain about our estimate. This can be done by using the normal distribution that are narrower than t-distribution and gives a narrower interval.

c.)

```
#c.)
df = n-1 #degrees of freedom
chi_l = st.chi2.ppf(1-a/2,df) #critical
chi_u  = st.chi2.ppf(a/2,df) #critical
l,u = ((n-1)*v)/chi_l, ((n-1)*v)/chi_u #lower and upper interval

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

By splitting the dataset into a training set and test set when doing machine learning it's possible to evaluate the performance of a ML algorithm. The procedure divides a dataset into two subsets. The second subset is put aside and not used to train the model. This second subset is referred to as the test dataset. The first subset, the training set, is used by providing input elements of the dataset to the model and then predictions are made and compared to the expected values. In short, the training dataset is used to fit the machine learning model, and the test dataset is used to evaluate the fit of the machine learning model. This procedure is best practiced when there is a sufficiently large dataset available. The goal is to estimate the performance of the ML model on new data.

Training error is the prediction error of the training set, while test error is the prediction error of the test set. The training error can underestimate the test error which is why its important to have an independent dataset to evaluate the model performance.

b.)
Typically, a ML algorithm include a parameter adjusting the complexity of the model. The goal is to find the model complexity that gives the smallest test error. This is found by e.g. constructing "validation sets" from your training sets (e.g. cross-validation, bootstrap).

5. Time series analysis and Fourier transformation

a.)
There are several tests that can be performed to test if there is a significant trend in Xt. Some examples are the OLS test, the run test or the Mann-Kendall test.

The Mann-Kendall test uses the raw data to detect possible trends. The test is based on the test statistics S defined as:

$$S = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \operatorname{sgn}(x_j - x_i)$$

Where xj is the sequential data values, n is the length of the dataset and sgn is:

$$\operatorname{sgn}(\theta) = \begin{cases} 1 & \text{if } \theta > 0 \\ 0 & \text{if } \theta = 0 \\ -1 & \text{if } \theta < 0 \end{cases}$$

The test calculates the test statistics and compares the value to a value read from a standard normal distribution table. A chosen alpha is used as the significance level of the test.

The M-K test is a non-parametric test which means it works for all distributions. This test can be used to find the trend with small sample sizes, however the more datapoints there is, the more likely it is to find a true trend.

b.)
Figure A shows the Fourier transform of Xt.
It's not C due to the lack of noise and it´s not B due to the inconsistent pattern between 0.4 and 0.5 frequency (the sum of amplitude).
The Fourier transform is a tool that breaks signals into alternate representations, characterized by sine and cosines. The Fourier transform shows that any signal can be rewritten as the sum of sinusoidal functions.