Examination

Notes

- For all words which are underlined[1], a translation to French is provided at the bottom of the page.
- You must not write in red, pencil[2] or erasable ballpen[3].
- Only the answers written on the provided squared paper with an EPFL logo will be marked. Anything written on draft sheets[4] or on the question sheet will not be graded.
- Write you name and Camipro SCIPER number in the respective fields on the first page of the provided EPFL paper.
- Additional EPFL paper can be provided and must be marked with your name and Camipro SCIPER number at the top. If you use extra paper you must have it stapled[5] together with your other paper by an exam invigilator[6] before handing it in.
- For each question, the computation[7] leading to the final result should be clear. Partial points can be given even if the final result is not correct.
- You can round *final* results to two significant digits[8] (for example, 35% or 7.2) or use fractions (for example 1/6).
- The term "mean[9]" always corresponds to the "arithmetic mean" here.
- All required statistical tables are provided on the pages after the questions.
- The maximum number of points awarded for each question is noted as, for example, [2p].

[1]souligné [2]crayon a papier [3]stylo effacable [4]papier broullion [5]agrafé [6]surveillant [7]calculus
[8]chiffres significatifs [9]moyenne

1 Superconducting crystals

You create a new method for producing superconducting crystals. You find that 20% of the crystals have superconducting critical temperatures above 10K (we call this probability $P(A) = 0.2$). Of those crystals, 60% are paramagnetic at room temperature (we call this probability $P(B|A) = 0.6$). Furthermore[10], 72% of the crystals are non-paramagnetic and do not have a superconducting critical temperature above 10k, $P(\bar{A} \cap \bar{B}) = 0.72$

{1p} a) What is the probability for a randomly chosen crystal to have a superconducting critical temperature above 10K and not be paramagnetic at room temperature, $P(A \cap \bar{B})$?

{2p} b) Create a probability tree diagram, including the values for $P(A)$, $P(\bar{A})$, $P(B|A)$,$P(\bar{B}|A)$,$P(B|\bar{A})$ and $P(\bar{B}|\bar{A})$

{2p} c) Given that a randomly chosen crystal is paramagnetic, what is the probability that it has a superconducting critical temperature above 10K?

Having produced a very large number of these crystals, you find that their resistivity (always measured in units of nΩm) is well-described by a normal/Gaussian distribution with a mean of 150 and a standard deviation[11] of 10.

{1p} d) What is the probability for a randomly chosen crystal to have a resistivity below 160?

{2p} e) What is the probability for a randomly chosen crystal to have a resistivity between 130 and 140?

{2p} f) What is the probability for the *mean* resistivity of a sample of 25 randomly chosen crystal to be below 155?

In a new research project, you develop another method to grow a new type of crystals. Your aim is to show that this new method produces crystals with a mean superconducting critical temperature that is larger than 20K. You produce 4 of those crystals and find critical temperatures (measured in K) of $[21, 19, 22, 26]$.

{2p} g) Compute the mean and the unbiased[12] estimator for the variance from this data.

{2p} h) Which type of statistical test would be suitable to show if you were successful?

{4p} i) Perform this test, using a confidence level[13] of 99% (significance level[14] of 1%)

2 5-sided die

You are studying a 5-sided die[15], with possible outcomes numbered 1,2,3,4,5. Initially, assume that all results have equal probability (i.e.[16] it is a fair die).

{1p} a) What is the probability of the outcomes "odd[17] number" and "number larger than 3"?

{2p} b) Are these two probabilities independent? Prove your answer any way you prefer.

{1p} c) What is the expectation value (for the mean) of this die?

{2p} d) In a game where playing once costs 10CHF, and the prize in one round corresponds to the thrown[18] number squared, calculate the expectation value for the overall gain or loss per round.

[10]de plus [11]ecart type [12]non-biasé [13]niveau de signification [14]niveau de signification [15]dé [16]c'est-à-dire [17]impaire [18]lancé

Now you try to build such a die, but you do not know if it is fair.

{2p} e) After throwing your die 5 times, you get the following results: [2,5,2,5,1]. Compute the mean, median and the unbiased estimator of the variance of this sample.

After throwing 200 times, the 5 possible results occur the following number of times:

Result	1	2	3	4	5
Number of occurences	48	35	38	33	46

{1p} f) Compute the difference between this outcome and the expectation values for each result for a fair die.

{1p} g) To test the hypothesis "this die is fair, each result has the same probability", which statistical test is suitable?

{4p} h) Perform this test, using a confidence level of 95% (significance level of 5%).

{1p} i) Was throwing 200 times sufficient to perform this test? Justify your statement.

A junior colleague of yours has written the following Python script to find the median of the data given in (e):

```
import numpy as np

def findMedian(inputArray):
    sortedArray = np.sort(inputArray)
    print("Sorted Array:")
    print(sortedArray)
    arrayLength = len(sortedArray)
    print("Length of the array:")
    print(arrayLength)
    medianPositionFloat = arrayLength/2+0.5
    medianPosition = int(medianPositionFloat)#transform to integer
    print("Position of the median:")
    print(medianPosition)
    median = sortedArray[medianPosition]
    return median

myArray = np.array([2,5,2,5,1])
myMedian = findMedian(myArray)
print("The median:")
print(myMedian)
```

{3p} j) Which outputs do you expect from this code? *Note that* `"print"` *calls inside a function will also produce an output.*

{2p} k) Why does the script give the wrong median? How can you correct it by making only a small change to the code above?

{3p} l) Furthermore, this code (also with the small correction) would not work for input arrays with an even[19] number of elements. Describe in words or with code (precise syntax is not important) how you would write a script that can also process inputs with an even number of elements. *There are several possible solutions, some longer, some shorter, every solution that works correctly for both even and odd numbers of elements is acceptable.*

[19] pair

1. {2p} [m)] In general, if the mean and the median of a set of data *are not* the same, this implies that the data *are not* symmetric around the mean. If the mean and the median of a set of data *are* the same, does that *always* imply that the data *is* symmetric around the mean? Prove your answer any way you like.

3 Thermal conductivities

You measure the thermal conductivity of two types of metals, Platinum (Pt) and Molybdenum (Mo). You can call the "type of metal" factor I. You measure them both at room temperature ("RT") and at liquid nitrogen temperature ("LN"). You can call the "temperature" factor J. Each possible configuration is measured 11 times. You find the following means and unbiased estimators for the standard deviation for your measurements. All thermal conductivities are given in units of W/m/K:

	$\bar{X}_{i,j,\bullet}$	$s_{i,j}$	N_S
Pt,RT	$\bar{X}_{1,1,\bullet} = 67$	$s_{1,1} = 11$	11
Pt,LN	$\bar{X}_{1,2,\bullet} = 83$	$s_{1,2} = 8$	11
Mo,RT	$\bar{X}_{2,1,\bullet} = 139$	$s_{2,1} = 14$	11
Mo,LN	$\bar{X}_{2,2,\bullet} = 211$	$s_{2,2} = 17$	11

{2p}a) Compute the partial/marginal means[20] for both factors and the total mean, and arrange[21] all means in a "table of means."

{2p} b) Compute the sum of the squared errors ($SS_{E,i,j}$) for each group and hence the total SS_E.

{1p} c) Compute the sum of the square difference between levels for each factor ($SS_{B,I}$ and $SS_{B,J}$).

{1p} d) Statistically speaking, which factor is the dominant one?

{5p} e) Create a complete two-factor ANOVA table *including interactions.* (Also include the Fisher statistic for each factor and for the interaction).

{2p} f) Test the null hypothesis "there is no interaction between the factors I and J" using a level of significance of 5%.

Now you focus on studying the dependence of the thermal conductivity, κ, of Platinum as a function of temperature T. You want to use linear regression to compare your data to different models.

{3p} g) For a model given by

$$\kappa = a + b_1 T$$

prove that the estimator for a, which minimizes the sum of the squared errors (SS_E) between the data and the model is given by $\hat{a} = \bar{\kappa} - \hat{b}_1 \bar{T}$ where $\hat{b}_1$ denotes the estimator for b_1

You have now taken data at 7 different temperatures. You fit your data with both models. The first contains only an offset and a linear slope (as above), the second additionally contains a term which is quadratic in temperature and has a pre-factor called b_2. You find the following SS_E for the two models:

Model	SS_E
$\kappa = a + b_1 T$	1255
$\kappa = a + b_1 T + b_2 T^2$	405

[20] moyenne partielle [21] trier

{1p} h) How many degrees of freedom does the error have in each model?

{2p} i) Is the second model statistically significantly better than the first model? Use a level of significance of 5%.

You have now automated your data acquisition with Python and successfully written a code that creates a numpy array called `measuredData`, where each element corresponds to a thermal conductivity.

{1p} j) However, you have to take a calibration factor into account, each measurement needs to be multiplied by 1.2. Which line of code would you write to produce an array called `calibratedData` that now contains data with the correct calibration?

Occasionally your measurement device malfunctions because of electrical spikes, and then produces erroneous outputs which give values larger than 1000. You have performed a measurement and created an array called `calibratedData` with the values $[68.4, 72.1, 1200.2, 80.4]$.

{1p} k) What output do you expect for the following line of code?

```
print(calibratedData[0:2])
```

{2p} l) Which line of code could you write to generate an output that lets you see (even for a large array) if your measurement contains an erroneous element (i.e. a value larger than 1000)? (*There are several good options.*)

{2p} m) To filter out such values and replace them by a `np.nan` ("not a number") value, your colleague has written the following function:

```
def myFilter(inputData):
    if inputData>1000.:
        return np.nan
    else:
        return inputData
```

You try to apply this function to your array by writing:

```
print(myFilter(calibratedData))
```

and you receive the following error message:

```
ValueError: The truth value of an array with more than one element is
ambiguous. Use a.any() or a.all()
```

How can you solve this issue so that you filter your data as desired? (*There are several possible solutions, and you are not forced to use your colleagues' function*).

Cumulative distribution function (CDF) of the centered and reduced normal distribution (z-table) / Tableau de la fonction de répartition normale centree et reduite.

z	$\Phi(z)$	z	$\Phi(z)$	z	$\Phi(z)$	z	$\Phi(z)$	z	$\Phi(z)$	z	$\Phi(z)$
0,00	0,500	0,72	0,764	1,44	0,9251	2,16	0,9846	2,88	0,99801	3,80	0,9999277
0,02	0,508	0,74	0,770	1,46	0,9279	2,18	0,9854	2,90	0,99813	3,84	0,9999385
0,04	0,516	0,76	0,776	1,48	0,9306	2,20	0,9861	2,92	0,99825	3,88	0,9999478
0,06	0,524	0,78	0,782	1,50	0,9332	2,22	0,9868	2,94	0,99836	3,92	0,9999557
0,08	0,532	0,80	0,788	1,52	0,9357	2,24	0,9875	2,96	0,99846	3,96	0,9999625
0,10	0,540	0,82	0,794	1,54	0,9382	2,26	0,9881	2,98	0,99856	4,00	0,9999683
0,12	0,548	0,84	0,800	1,56	0,9406	2,28	0,9887	3,00	0,99865	4,04	0,9999733
0,14	0,556	0,86	0,805	1,58	0,9429	2,30	0,9893	3,02	0,99874	4,08	0,9999775
0,16	0,564	0,88	0,811	1,60	0,9452	2,32	0,9898	3,04	0,99882	4,12	0,9999811
0,18	0,571	0,90	0,816	1,62	0,9474	2,34	0,9904	3,06	0,99889	4,16	0,9999841
0,20	0,579	0,92	0,821	1,64	0,9495	2,36	0,9909	3,08	0,99996	4,20	0,9999867
0,22	0,587	0,94	0,826	1,66	0,9515	2,38	0,9913	3,10	0,99903	4,24	0,9999888
0,24	0,595	0,96	0,831	1,68	0,9535	2,40	0,9918	3,12	0,99910	4,28	0,9999907
0,26	0,603	0,98	0,836	1,70	0,9554	2,42	0,9922	3,14	0,99916	4,32	0,9999922
0,28	0,610	1,00	0,841	1,72	0,9573	2,44	0,9927	3,16	0,99921	4,36	0,9999935
0,30	0,618	1,02	0,846	1,74	0,9591	2,46	0,9931	3,18	0,99926	4,40	0,9999946
0,32	0,626	1,04	0,851	1,76	0,9608	2,48	0,9934	3,20	0,99931	4,44	0,9999955
0,34	0,633	1,06	0,855	1,78	0,9625	2,50	0,9938	3,22	0,99936	4,48	0,9999963
0,36	0,641	1,08	0,860	1,80	0,9641	2,52	0,9941	3,24	0,99940	4,52	0,9999969
0,38	0,648	1,10	0,864	1,82	0,9656	2,54	0,9945	3,26	0,99944	4,56	0,9999974
0,40	0,655	1,12	0,869	1,84	0,9671	2,56	0,9948	3,28	0,99948	4,60	0,9999979
0,42	0,663	1,14	0,873	1,86	0,9686	2,58	0,9951	3,30	0,99952	4,64	0,9999983
0,44	0,670	1,16	0,877	1,88	0,9799	2,60	0,9953	3,32	0,99955	4,68	0,9999986
0,46	0,677	1,18	0,881	1,90	0,9713	2,62	0,9956	3,34	0,99958	4,72	0,9999988
0,48	0,684	1,20	0,885	1,92	0,9726	2,64	0,9959	3,36	0,99961	4,76	0,9999990
0,50	0,691	1,22	0,889	1,94	0,9738	2,66	0,9961	3,38	0,99964	4,80	0,9999992
0,52	0,698	1,24	0,893	1,96	0,9750	2,68	0,9963	3,40	0,99966	4,84	0,9999994
0,54	0,705	1,26	0,896	1,98	0,9761	2,70	0,9965	3,42	0,99969	4,88	0,9999995
0,56	0,712	1,28	0,900	2,00	0,9772	2,72	0,9967	3,44	0,99971	4,92	0,9999996
0,58	0,719	1,30	0,903	2,02	0,9783	2,74	0,9969	3,46	0,99973	4,96	0,9999996
0,60	0,726	1,32	0,907	2,04	0,9793	2,76	0,9971	3,48	0,99975	5,00	0,9999997
0,62	0,732	1,34	0,910	2,06	0,9803	2,78	0,9973	3,50	0,99977	5,04	0,9999998
0,64	0,739	1,36	0,913	2,08	0,9812	2,80	0,9974	3,52	0,99978	5,08	0,9999998
0,66	0,745	1,38	0,916	2,10	0,9821	2,82	0,9976	3,54	0,99980	5,12	0,9999998
0,68	0,752	1,40	0,919	2,12	0,9830	2,84	0,9977	3,56	0,99981	5,16	0,9999999
0,70	0,758	1,42	0,922	2,14	0,9838	2,86	0,9979	3,58	0,99983	5,20	0,9999999

Quantiles of the Student t distribution / Quantiles de la loi t de Student
ν =df

ν	$qt_\nu(95\%)$	$qt_\nu(97{,}5\%)$	$qt_\nu(99\%)$	ν	$qt_\nu(95\%)$	$qt_\nu(97{,}5\%)$	$qt_\nu(99\%)$
1	6,314	12,71	31,82	21	1,721	2,080	2,518
2	2,920	4,303	6,965	22	1,717	2,074	2,508
3	2,353	3,182	4,541	23	1,714	2,069	2,500
4	2,132	2,776	3,747	24	1,711	2,064	2,492
5	2,015	2,571	3,365	25	1,708	2,060	2,485
6	1,943	2,447	3,143	26	1,706	2,056	2,479
7	1,895	2,365	2,998	27	1,703	2,052	2,473
8	1,860	2,306	2,896	28	1,701	2,048	2,467
9	1,833	2,262	2,821	29	1,699	2,045	2,462
10	1,812	2,228	2,764	30	1,697	2,042	2,457
11	1,796	2,201	2,718	32	1,694	2,037	2,449
12	1,782	2,179	2,681	34	1,691	2,032	2,441
13	1,771	2,160	2,650	36	1,688	2,028	2,434
14	1,761	2,145	2,624	38	1,686	2,024	2,429
15	1,753	2,131	2,602	40	1,684	2,021	2,423
16	1,746	2,120	2,583	50	1,676	2,009	2,403
17	1,740	2,110	2,567	60	1,671	2,000	2,390
18	1,734	2,101	2,552	120	1,658	1,980	2,358
19	1,729	2,093	2,539	∞	1,645	1,960	2,326
20	1,725	2,086	2,528				

Pour ν suffisamment grand, on peut substituer aux quantiles de la loi t les quantiles de la loi normale.
For sufficiently large ν ,the quantiles become the same as those of the normal distribution.

Chi–2 (chi–square) table/Quantiles de la loi khi–deux

$\nu = df$

ν	$q\chi^2_\nu(95\%)$	$q\chi^2_\nu(97{,}5\%)$	$q\chi^2_\nu(99\%)$
1	3,841	5,024	6,635
2	5,991	7,378	9,210
3	7,815	9,348	11,34
4	9,488	11,14	13,28
5	11,07	12,83	15,09
6	12,59	14,45	16,81
7	14,07	16,01	18,48
8	15,51	17,53	20,09
9	16,92	19,02	21,67
10	18,31	20,48	23,21
11	19,68	21,92	24,72
12	21,03	23,34	26,22
13	22,36	24,74	27,69
14	23,68	26,12	29,14
15	25,00	27,49	30,58
16	26,30	28,85	32,00
17	27,59	30,19	33,41
18	28,87	31,53	34,81
19	30,14	32,85	36,19
20	31,41	34,17	37,57
21	32,67	35,48	38,93
22	33,92	36,78	40,29
23	35,17	38,08	41,64
24	36,42	39,36	42,98
25	37,65	40,65	44,31
26	38,89	41,92	45,64
27	40,11	43,19	46,96
28	41,34	44,46	48,28
29	42,56	45,72	49,59
30	43,77	46,98	50,89
32	46,19	49,48	53,49
34	48,60	51,97	56,06
36	51,00	54,44	58,62
38	53,38	56,90	61,16
40	55,76	59,34	63,69
50	67,50	71,42	76,15
60	79,08	83,30	88,38
70	90,53	95,02	100,4
80	101,9	106,6	112,3
90	113,1	118,1	124,1
100	124,3	129,6	135,8

95% Quantiles of the Fisher law (F-table)/
95% Quantiles de la loi F_{ν_1,ν_2} de Fisher

	$\nu_1 = 1$	2	3	4	5	6	7	8	10	12	24	∞
$\nu_2 = 1$	161,4	199,5	215,7	224,6	230,2	234,0	236,8	238,9	241,9	243,9	249,1	254,3
2	18,51	19,00	19,16	19,25	19,30	19,33	19,35	19,37	19,40	19,41	19,45	19,50
3	10,13	9,552	9,277	9,117	9,013	8,941	8,887	8,845	8,786	8,745	8,639	8,526
4	7,709	6,944	6,591	6,388	6,256	6,163	6,094	6,041	5,964	5,912	5,774	5,628
5	6,608	5,786	5,409	5,192	5,050	4,950	4,876	4,818	4,735	4,678	4,527	4,365
6	5,987	5,143	4,757	4,534	4,387	4,284	4,207	4,147	4,060	4,000	3,841	3,669
7	5,591	4,737	4,347	4,120	3,972	3,866	3,787	3,726	3,637	3,575	3,410	3,230
8	5,318	4,459	4,066	3,838	3,687	3,581	3,500	3,438	3,347	3,284	3,115	3,928
9	5,117	4,256	3,863	3,633	3,482	3,374	3,293	3,230	3,137	3,073	2,900	2,707
10	4,965	4,103	3,708	3,478	3,326	3,217	3,135	3,072	2,978	2,913	2,737	2,538
11	4,844	3,982	3,587	3,357	3,204	3,095	3,012	2,948	2,854	2,788	2,609	2,404
12	4,747	3,885	3,490	3,259	3,106	2,996	2,913	2,849	2,753	2,687	2,505	2,296
13	4,667	3,806	3,411	3,179	3,025	2,915	2,832	2,767	2,671	2,604	2,420	2,206
14	4,600	3,739	3,344	3,112	2,958	2,848	2,764	2,699	2,602	2,534	2,349	2,131
15	4,543	3,682	3,287	3,056	2,901	2,790	2,707	2,641	2,544	2,475	2,288	2,066
16	4,494	3,634	3,239	3,007	2,852	2,741	2,657	2,591	2,494	2,425	2,235	2,010
17	4,451	3,592	3,197	2,965	2,810	2,699	2,614	2,548	2,450	2,381	2,190	1,960
18	4,414	3,555	3,160	2,928	2,773	2,661	2,577	2,510	2,412	2,342	2,150	1,917
19	4,381	3,522	3,127	2,895	2,740	2,628	2,544	2,477	2,378	2,308	2,114	1,878
20	4,351	3,493	3,098	2,866	2,711	2,599	2,514	2,447	2,348	2,278	2,082	1,843
21	4,325	3,467	3,072	2,840	2,685	2,573	2,488	2,420	2,321	2,250	2,054	1,812
22	4,301	3,443	3,049	2,817	2,661	2,549	2,464	2,397	2,297	2,226	2,028	1,783
23	4,279	3,422	3,028	2,796	2,640	2,528	2,442	2,375	2,275	2,204	2,005	1,757
24	4,260	3,403	3,009	2,776	2,621	2,508	2,423	2,355	2,255	2,183	1,984	1,733
25	4,242	3,385	2,991	2,759	2,603	2,490	2,405	2,337	2,236	2,165	1,964	1,711
26	4,225	3,369	2,975	2,743	2,587	2,474	2,388	2,321	2,220	2,148	1,946	1,691
27	4,210	3,354	2,960	2,728	2,572	2,459	2,373	2,305	2,204	2,132	1,930	1,672
28	4,196	3,340	2,947	2,714	2,558	2,445	2,359	2,291	2,190	2,118	1,915	1,654
29	4,183	3,328	2,934	2,701	2,545	2,432	2,346	2,278	2,177	2,104	1,901	1,638
30	4,171	3,316	2,922	2,690	2,534	2,421	2,334	2,266	2,165	2,092	1,887	1,622
32	4,149	3,295	2,901	2,668	2,512	2,399	2,313	2,244	2,142	2,070	1,864	1,594
34	4,130	3,276	2,883	2,650	2,494	2,380	2,294	2,225	2,123	2,050	1,843	1,569
36	4,113	3,259	2,866	2,634	2,477	2,364	2,277	2,209	2,106	2,033	1,824	1,547
38	4,098	3,245	2,852	2,619	2,463	2,349	2,262	2,194	2,091	2,017	1,808	1,527
40	4,085	3,232	2,839	2,606	2,449	2,336	2,249	2,180	2,077	2,003	1,793	1,509
60	4,001	3,150	2,758	2,525	2,368	2,254	2,167	2,097	1,993	1,917	1,700	1,389
120	3,920	3,072	2,680	2,447	2,290	2,175	2,087	2,016	1,910	1,834	1,608	1,254
∞	3,841	2,996	2,605	2,372	2,214	2,099	2,010	1,938	1,831	1,752	1,517	1,000