UFR de mathématique et d'informatique Université de Strasbourg

Project report M1:

SCHUFFENECKER Antoine
Roll: *Intern*
Class: *Master 1 Csmi*
Session: *2020/2021*
Email: *antoine.schuffenecker@gmail.com*

Course: *PROJET* – Teacher: *Cristhophe PRUD'HOMME*
Submission date: *29/05*

Contents

Introduction

1.Context:

The sport has a great influence on health, it is shown in many article . For example in the study of the young population in Belgium, it is shown and demonstrated that a healthy and reasonable practice of sports in our daily life habits reduces the risk of cardiac incident. The earlier these life habits are introduced in the individuals routine the greater the positive impact is. This is a public health organisation problem because like the study mentioned before we can fin other ones which states on the impact of physical activities on risks of diseases such as cancer, diabetes,etc.
Now we want to try to evaluate how the population, in terms of physical abilities, reacted to the Covid-19 which forced a lot of people to stop any sports.
The students of STAPS takes physical tests when they arrive the first year in the cursus. These tests are organised around 4 core notions : Strength, Speed, Coordination and Endurance. The results of these test gives a grade. The common trend is to say that performances are worse every year which might not be true.

However the Covid happened and with it a lot of students could not practice any sport for long periods of time. So we might see an impact on performances.
There is at the moment one data set:

- The whole results of all STAPS students since 1999, roughly 20000 results.

2.Supervisors:

My supervisor is Christophe Schnitzler. Expert in physical mobility, formation of teachers and Education to healthcare by sport.

3.Subject:

With the context given we have now 3 major questions :

- How did the overall performances of all students changes during all these years ?
- How exactly did the covid impacted the students on a physical level ?
- Did the covid impacted the students differently regarding their gender ?

I) Data

1.Presentation:

Since 1999 the STAPS direction imposes on their students to take the BCPE test which consists of several physical exams to determine the physical shape of their students. These are separated in categories speed, endurance, strength, coordination and others.
All this data is stored in files. Since some modalities or responsible have changed through the years, the files are not exploitable with no pre-treatments to gather all of it in an efficient way. For example some of the trials have changed to measure with a better accuracy physical qualities. The first batch of data is dated between 1999 and 2015, the second one is 2016,the data from 2017 and 2018 are not exploitable, it is missing way too many results about the exams, then each year is a single file but has a similar architecture. The data stops in 2021.
There is some precision about the data, on the 1st batch it is an excel file, each sheet represents a year. In 1999 there are no results for mens, and in 2015 the grade of the overall test is missing. On the second batch it is a pdf file which needs special treatment. On the last one it is again different excel files with different sheets but on the results one are used.

2.Data˙Cleaning:

The main work that I had to do was to clean all the data. My results come from a lot of trials to be sure that every errors and exceptions are treated to obtain a very clean dataframe. Which will be the easiest to use. I will explain all my methodology for each batch.

To obtain useful data we need it to be present in most years with little to no change on the exercises asked to the students so we can see clearly the evolution and the impact of any perturbation like we would like to see.

The list I chose is the following:

```
liste=['L1','Année','Allure','Vit. 20m','Vit. 50m','Développer\nCouché','Détente\nVerticale','Souplesse','Equilibre','Coordination','50M ','BCPE\n
liste2=['Allure','Vit. 20m','Vit. 50m','Développer\nCouché','Détente\nVerticale','Souplesse','Equilibre','Coordination','50M ','BCPE\n']
```

('L1' and 'Année' are here to give the gender and the year of the results of a line).

On the first hand for the first batch, I started with moving the following data from excel in a dataframe.

NOmS	Année	sexe	VmA	V20m	V30m	V50m	dvt couché	détente vert	abdos 40s	coord (sec)
AVALLONE FABIO	2015	m	19,5	23,8	31,5	27,9	80	43,4	42	15,9
CHEVRIER Lucien	2015	m	19	25,1	32,1	28,9	65	44,7	41	15,32
RIEMER léo	2015	m	19	23,9	31,5	28,0	77,5	41,5	37	15,9
HERMANN timothée	2015	m	19	22,9	29,3	26,4	72,5	34,4	36	16,44
PARISSE jason	2015	m	19	24,9	30,8	28,1	87,5	45,9	45	15,7
BORIS Simon	2015	m	19	24,2	31,0	27,9	65	43,8	40	15,1
CRIQUI Guillaume	2015	m	19	23,2	29,5	26,6	60	43,4	38	15,66
SCHNEIDER lucas	2015	m	19	24,8	32,6	29,0	60	44	42	16,16
STAUB benjamin	2015	m	19	23,8	29,7	27,0	72,5	48,2	46	15,84
BENOMAR Rémy	2015	m	19	24,7	31,6	28,4	47,5	38,7	37	15,65
WEBER Edouard	2015	m	18,5	23,5	30,5	27,3	67,5	45,5	52	15,28
MEHN Vincent	2015	m	18,5	22,1	27,1	24,9	60	34,7	39	16,07
LAMBERT Valentin	2015	m	18,5	24,7	30,5	27,9	55	50,8	44	15,91
MASCHINO Quentin	2015	m	18,5	24,7	30,9	28,1	87,5	33,5	43	16,1
WACK xavier	2015	m	18,5	23,8	29,6	26,9	60	40,7	34	16,62
BRUN VALENTIN	2015	m	18,5	24,6	31,5	28,3	62,5	49,4	46	14,91
WEHR Hugo	2015	m	18,5	23,5	30,4	27,2	80	38	50	15,62
SCHULLER axel	2015	m	18,5	24,8	31,8	28,6	60	44,9	40	15,09
RINGLER tristan	2015	m	18,5	23,8	31,4	27,9	85	39,8	46	15,78
LECOUSTRE maxime	2015	m	18,5	24,4	30,3	27,6	45	33,8	38	15,15
FAUVIAUX Mickael	2015	m	18,5	24,1	30,4	27,5	60	35,3	42	15,62
JAEGER Jérémy	2015	m	18,5	23,8	32,4	28,3	80	41,4	46	14,94
MATHIS Robin	2015	m	18,5	24,3	32,0	28,4	87,5	43,4	51	15,81
MULLER lucas	2015	m	18	24,0	30,9	27,7	75	41,7	42	14,94
RITTER romain	2015	m	18	24,2	31,1	27,9	65	40,4	37	15,59
DENIER Alex	2015	m	18	24,5	30,5	27,8	100	43	37	15,07
FEDERMANN Théo	2015	m	18	24,1	28,6	26,6	82,5	49	45	15,1
NOUMBA YAMANGAM Michael	2015	m	18	24,8	31,2	28,3	120	51,3	51	16,43
CENDIER Ruddy	2015	m	18	24,1	30,7	27,6	92,5	41	39	16,34

BCPE_ens | 2015 | 2014 | 2013 | 2012 | 2010 | 2009_ens | 2008 | 2007 | 2006 | 2005 | 2004 | 2003 | 2002 | ...

I had to replace every ‘m’,’g’ and some variations (some with space in it) with ‘M’, which will help later display all the data. The same had to be done for female students but with only variation of ‘f’ replaced by ‘F’. Then since the columns had not the same name through the years I chose the ones who were used the most recently. To do so i had to copy the right columns and then select only the ones that will be needed later

```
df_tot=pd.read_excel(xls, 'BCPE_ens')
df_tot['sexe']=df_tot['sexe'].replace('m','M')
df_tot['sexe']=df_tot['sexe'].replace('m ','M')
df_tot['sexe']=df_tot['sexe'].replace('G','M')
df_tot['sexe']=df_tot['sexe'].replace('g','M')
df_tot['sexe']=df_tot['sexe'].replace('f','F')
df_tot['sexe']=df_tot['sexe'].replace('f ','F')

df_tot['Allure']=df_tot['VmA']
df_tot['L1']=df_tot['sexe']
df_tot['Vit. 20m']=df_tot['V20m']
df_tot['Vit. 50m']=df_tot['V50m']
df_tot['Développer\nCouché']=df_tot['dvt couché']
df_tot['Détente\nVerticale']=df_tot['détente vert']
df_tot['50M ']=df_tot['V50m NL']
df_tot['Souplesse']=df_tot['soupl (cm)']
df_tot['Coordination']=df_tot['coord (sec)']
df_tot['Equilibre']=df_tot['equil (nb chute)']
df_tot['BCPE\n']=df_tot['Note /20']
df_tot=df_tot[liste]
```

Then on the second batch I had to use a different approach because it is a pdf file, the mains issue with the file is that the pdf was not in a exploitable format, I had to search by hand to reassign the right columns to the variable needed. Since the format was not well design a lot of the data was very hard to exploit. It might later gives some issue with the 2016 year. Furthermore the way of writing float number with ”,” introduced a lot of bugs that I had a hard time to detect.

		MOY /20	ENDURANCE		VITESSE					FORCE					
			Palier KM/H	NOTE/20	20M en sec	Note /10	50M en sec	Note /10	NOTE/20	Dév. couché en Kg	Note /10	Détente en cm	Note/5	Gainage en min	N
AURELIE	F	11,2	15	8	3,69	6,5	8,14	4	10,5	37	6,5	27	2	3,09	
LAURYN	F	3,8	DISP	0	4,02	4	9,38	0	4	42	7,5	21,8	1	1,15	
PIERRE	M	9	17	8	3,21	8,5	6,95	5,5	14	67	6	36,4	1	3	
DORIAN	M	12	18	10	3,34	7	7,14	5	12	87	8	37,8	1	4	
MAXIME	M	8,8	17	8	3,46	6	7,6	2,5	8,5	62	5,5	41,8	1,5	2,1	
MADISON	F	8,2	13	4	3,71	6,5	8	4,5	11	33	5,5	24,7	1,5	2,3	
VINCENT	M	DEF	abs	#N/A	3,2	8,5	6,9	5,5	14	76	7	41,5	1,5	4	
JULIEN	M	12,1	16,5	7	3,14	9	6,65	6,5	15,5	81	7,5	45,9	2	4	
FELIX	M	10,7	17	8	3,17	8,5	6,78	6	14,5	55	4	40,2	1,5	3,03	
CORALINE	F	9,6	13,5	5	3,73	6	7,14	7,5	13,5	26	4	32,3	2,5	2,12	
NAWALIATOU	F	10,3	15,5	9	3,55	7,5	7,6	6	13,5	45	8	34,8	2,5	3,01	
NICOLAS	M	13,4	18	10	3,02	10	6,42	7,5	17,5	80	7	44,1	2	4	
YLIES	M	13,1	20	14	3,25	8	6,8	6	14	72	6,5	41,2	1,5	3,36	
ALEXANDRE	M	9,9	15	4	3,21	8,5	6,83	6	14,5	85	7,5	43,4	2	2,27	
THOMAS	M	HN	HN	#N/A	HN	#N/A	HN	#N/A	#N/A	HN	#N/A	HN	#N/A	HN	#
LUCAS	M	13,1	18	10	3,21	8,5	6,83	6	14,5	80	7	46,1	2	3	
SLIM	M	10	17,5	9	3,22	8	6,85	6	14	46	3,5	36,7	1	3	
NOEMIE	F	14,1	15	8	3,5	8	7,62	5,5	13,5	53	8,5	40,4	3,5	3,21	
YANIS	M	10,4	16	6	3,21	8,5	6,84	6	14,5	85	7,5	35,6	1	3	
AYET SAFYA	F	10,8	15,5	9	3,48	8,5	7,5	6	14,5	35	6	31,5	2,5	2,12	
REDOUANE	M	7,2	16,5	7	3,3	7,5	7,19	4,5	12	79	7	34,7	0,5	1,4	
LUCIE	F	11,6	14,5	7	4	4	8,65	2	6	40	7	30,5	2	3,21	
AMANDINE	F	10,4	13,5	5	3,65	7	8,01	4,5	11,5	32	5,5	36,1	3	1,46	
LUCAS	M	9,6	17	8	3,26	8	7,23	4,5	12,5	60	5	38,2	1	2,37	
LOUIS	M	11,9	16	6	3,02	10	6,42	7,5	17,5	80	7	41,7	1,5	2,21	
KEVIN	M	11,6	16	6	3,15	9	6,8	6	15	100	8,5	43,4	2	3,31	
QUENTIN	M	8,5	18	10	3,13	9	6,77	6	15	45	3	39,1	1,5	2,2	
PASCAL	M	#####	16	6	3,16	8,5	6,79	6	14,5	53	4	36,3	1	4	
THEO	M	10,4	16	6	3,3	7,5	7,09	5	12,5	65	5,5	40,1	1,5	4,02	
LUKAS	M	10,6	16	6	3,24	8	6,84	6	14	58	4,5	40,8	1,5	2,2	
YANN	M	HN	abs	#N/A					#VALEUR!					HN	#
ALEXANDRE	M	#N/A	abs	#N/A	3,34	7	7,15	5	12						
TOM	M	#N/A	abs	#N/A					#VALEUR!						
ALEXY	M	8,7	14	2	3,33	7,5	7,16	5	12,5	60	5	33,8	0,5	2,3	
AXEL	M	9,5	15	4	3,34	7	7,22	4,5	11,5	53	4	35,3	1	3,01	

```
df1=tabula.read_pdf(r'C:\Users\antoi\OneDrive\Desktop\Projet_Staps\DECA pratique notes 2016.xls.pdf',pages='1')

df_p=df1[0]
df_p['Année']=2016
df_p['L1']=df_p['Unnamed: 5']
df_p['BCPE\n']=df_p['ENDURANCE']
df_p['Allure']=df_p['VITESSE']
df_p['Vit. 20m']=df_p['C.E.S']
df_p['Vit. 50m']=df_p['Unnamed: 7']
df_p['Développer\nCouché']=df_p['Unnamed: 10']
df_p['Détente\nVerticale']=df_p['Unnamed: 12']
df_p['Souplesse']=df_p['Unnamed: 21']
df_p['Equilibre']=df_p['Unnamed: 19']
df_p['Coordination']=df_p['Unnamed: 17']
df_p['50M ']=df_p['Unnamed: 24']
df_p=df_p[liste]
for i in liste2:
    df_p[i]=df_p[i].apply(str).str.replace(',','.')
```

For the latest batch I tought i would only have to create a a columns 'Année' but an error in the columns of the year 2020 made me redefine all the columns to ensure the right results. I also had to change the 'G' into 'M'. BUt apart from that the data was for the most part much more easier to use (it has to do with choice i made for the columns names but still a good point).

	TESTS BCPE 2			Temps	Allure	N/20	Endurance	Vit. 20m	N/20	Vit. 50m	N/20	Vitesse	Développer Couché	Poids	Ratio
M	21817022	ABDELKRIM	MOUSTAFA	24	21,5	20	20	3,33	15	6,96	10	12,5	65	74	0,88
M	21819964	ABOU EL HASSEN	ABDELKAR	21	20	18	18	3,11	18	6,64	12	15	58	63	0,92
M	21808085	ADDI	ILIES	21	20	18	18	3,19	17	6,89	11	14	52	70	0,74
M	21904455	AIME	FLORENT	19	19	16	16	DSP	DSP	DSP	DSP	DSP	81	69	1,17
M	21814491	AJAPUHNYA	RAPHAEL	11	15	8	8	3,19	17	6,89	11	14	73	98	0,74
M	21515935	ALLANGBA	KEVIN	18	18,5	15	15	3,02	20	6,44	14	17	81	63	1,29
F	21804356	ALLOUCHE	SARAH	13	16	13	13	3,6	15	7,85	10	12,5	37	63	0,59
M	21914069	AMELLOUK	MOHAMED	22	20,5	19	19	3,03	20	6,44	14	17	64	75	0,85
M	21905377	AMOUROUX	JULIAN	19	19	16	16	3,01	20	6,56	13	16,5	48	68	0,71
M	21906698	ANDRÉ	QUENTIN	19	19	16	16	3,24	16	7,05	9	12,5	52	64	0,81
M	21918194	ANDRIANIRINA	TOJO	16	17,5	13	13	3,06	19	6,82	11	15	46	71	0,65
M	21904955	ANDRIANTSIZAFY	FANIRY	13	16	10	10	3,4	14	7,49	6	10	51	94	0,54
M	21821525	ANGOUE NDOUTOUME	JEAN-FRAN	19	19	16	16	3,09	19	6,64	12	15,5	85	92	0,92
M	21905581	ANTOINE	MARIUS	22	20,5	19	19	3,03	20	6,53	13	16,5	48	68	0,71
M	21911419	ARBOGAST	ANTOINE	22	20,5	19	19	3,14	18	6,7	12	15	52	63	0,83
M	21907960	ARBOGAST	GUILLAUM	18	18,5	15	15	3,06	19	6,63	12	15,5	0	82	0,00
M	21906708	ATAOUIL	ILYASSE	20	19,5	17	17	2,92	20	6,15	16	18	67	67	1,00
M	21811011	ATES	MEHMET	19	19	16	16	3,23	16	6,79	11	13,5	58	49	1,18
M	21906231	ATTOU	MATHIEU	20	19,5	17	17	3,03	20	6,39	14	17	70	66	1,06
F	21903189	AVDIBEGOVIC	LETITIA	10	14,5	10	10	3,81	11	8,58	5	8	21,5	49	0,44
M	21909665	AYDIN	MANUEL	21	20	18	18	3,04	20	6,53	13	16,5	51	63	0,81
F	21801187	AYED	SABRINA	13	16	13	13	3,56	15	8,02	9	12	23,5	47	0,50
M	21814329	AZIZI	MOHAMED	14	16,5	11	11	3,2	17	6,78	11	14	75	65	1,15
F	21910133	BACHMEYER	LOLA	13	16	13	13	3,45	17	7,63	11	14	38	65	0,58
M	21907731	BAECHER	ADRIEN	22	20,5	19	19	2,85	20	6,04	17	18,5	71	76	0,93
M	21808624	BAECHTEL	LOÏC	18	18,5	15	15	3,31	15	6,96	10	12,5	70	63	1,11
F	21808640	BAILLY	LUCIE	8	13,5	8	8	3,7	13	8,34	6	9,5	46	66	0,70
F	21903048	BAMIDELE	MYRIAM	10	14,5	10	10	3,73	13	8,17	8	10,5	35	72	0,49
F	21902199	BARABANT	LUCIE	18	18,5	18	18	3,85	11	8,52	5	8	32,5	47	0,69
F	21909614	BARBIER	MANON	17	18	17	17	3,33	19	7,19	15	17	41	54	0,76
M	21908752	BARHDADI	RIYAD	11	15	8	8	ABI	0	ABI	0	0	DSP	117	0,00
M	21907593	BARRAU	NATHAN	20	19,5	17	17	3,08	19	6,65	12	15,5	46,5	66	0,70
M	21912859	BASCHUNG	ELI	18	18,5	15	15	3,2	17	6,99	10	13,5	60	58	1,03
M	21904032	BASTIDE	HUGO	21	20	18	18	3,05	19	6,34	14	16,5	85	64	1,33
M	21914415	BAUDRILLART	ELIOT	20	19,5	17	17	3,26	16	7,06	9	12,5	75	62	1,21
M	21813740	BAUMGARTNER	ARTHUR	15	17	12	12	3,19	17	6,71	12	14,5	81	73	1,11
M	21910724	BAYE	WILLIAM	17	18	14	14	3,06	19	6,62	12	15,5	52	73	0,71
F	21903075	BECK	LÉA	16	17,5	16	16	3,55	16	7,98	9	12,5	23,5	61	0,39
F	21917107	BEILE	SHANIA	10	14,5	10	10	3,52	16	8,11	8	12	35	68	0,51

APOGEE | Affichage | Podiums | **résultats** | Feuil1 | Notes Ecrit | Podiums 19-20 | Moins bons résultats | Natation

```
df1=df2019.copy()
df2=df2020.copy()
df3=df2021.copy()
df1['Année']=2019
df2['Année']=2020
df3['Année']=2021

df2.set_axis(['L1','TESTS BCPE 2019-2020','Unnamed: 2','Unnamed: 3', 'Temps', 'Allure','N /
'N /20.1','Vit. 50m','N /20.2','Vitesse',   'Développer\nCouché',  'Poids','Ratio','N /10',
 'N /10.1',  'Force', 'Coordination', 'N/10', 'Souplesse',  'N/5','Equilibre', 'N/5.1', 'Mo
'50M ','N /20.3','Natation','Tests BCPE\nPratique','Unnamed: 31','Place/','Tests BCPE\nThéo
'Unnamed: 34','Place/.1','BCPE\n','Année'], axis='columns', inplace=True)
df_temp=pd.concat([df1,df2,df3])
df_temp['L1']=df_temp['L1'].replace('G','M')
df_temp=df_temp[liste]
```

Now we just have to concatenate all these dataframe and we will treat it as a whole.

In the end there are still some rows which are unusable. I thought about which one to exclude, since we only care about the performances and the actual results in each trials any student who was absent to any of them will be discarded, for those who has already take the test any prior year will be also discarded (to only have one occurrence of those students), finally the one who are dispensed to do the trials are obviously also excluded of the study.

```
# Clean de toutes les épreuves
df=pd.concat([df_temp,df_tot,df_p])

df=df[(df['L1']=='M')|(df['L1']=='F')]
df=df[(df['L1']=='M')|(df['L1']=='F')]
df=df[(df['Allure']!='ABI')&(df['Allure']!='DSP')&(df['Allure']!='VAL')&(df['Allure']!='AB
df=df[(df['Vit. 20m']!='ABI')&(df['Vit. 20m']!='DSP')&(df['Vit. 20m']!='VAL')&(df['Vit. 20
df=df[(df['Développer\nCouché']!='ABI')&(df['Développer\nCouché']!='DSP')&(df['Développer\
df=df[(df['Détente\nVerticale']!='ABI')&(df['Détente\nVerticale']!='DSP')&(df['Détente\nVe
df=df[(df['Souplesse']!='ABI')&(df['Souplesse']!='DSP')&(df['Souplesse']!='VAL')&(df['Soup
df=df[(df['Equilibre']!='ABI')&(df['Equilibre']!='DSP')&(df['Equilibre']!='VAL')&(df['Equi
df=df[(df['Coordination']!='ABI')&(df['Coordination']!='DSP')&(df['Coordination']!='VAL')&
df=df[(df['50M ']!='ABI')&(df['50M ']!='DSP')&(df['50M ']!='VAL')&(df['50M ']!='ABJ')&(df[
df=df[(df['BCPE\n']!='DEF')]
df=df.replace(np.nan,0)
```

There is still some actions needed, change the data to be sure all trials over the year are measured with the same units, and exclude any strange rows.

```
df['Vit. 20m'] = df['Vit. 20m'].astype(float, errors = 'raise')
df['Vit. 50m'] = df['Vit. 50m'].astype(float, errors = 'raise')
df['Développer\nCouché'] = df['Développer\nCouché'].astype(float, errors = 'raise')
df['Détente\nVerticale'] = df['Détente\nVerticale'].astype(float, errors = 'raise')
df['Souplesse'] = df['Souplesse'].astype(float, errors = 'raise')
df['Equilibre'] = df['Equilibre'].astype(float, errors = 'raise')
df['Coordination'] = df['Coordination'].astype(float, errors = 'raise')
df['50M '] = df['50M '].astype(float, errors = 'raise')
```

4.Cleaning˙result:

So this is the result that I get

	L1	Année	Allure	Vit. 20m	Vit. 50m	Développer\nCouché	\
2	M	2019	21	3.39	6.59	62.0	
3	M	2019	14.5	3.98	7.32	34.0	
4	M	2019	18.5	3.53	6.69	58.0	
5	M	2019	16	4.32	7.90	44.0	
6	M	2019	17	3.95	7.20	75.5	
..	..	...	...	...	...	...	
65	M	2016	17	2.78	5.93	85.0	
66	M	2016	18	3.50	7.39	58.0	
67	M	2016	18	3.22	6.94	62.0	
69	F	2016	15.5	3.23	7.08	46.0	
70	M	2016	16.5	3.35	7.29	82.0	

	Détente\nVerticale	Souplesse	Equilibre	Coordination	50M	BCPE\n
2	43.0	-19.0	5.0	28.60	66.63	9.375
3	37.1	6.0	4.0	30.10	41.07	7.725
4	41.0	-2.0	10.0	29.80	46.36	9.9
5	24.3	-32.0	10.0	29.10	52.03	7.725
6	36.1	1.0	10.0	29.40	52.84	10.35
..	...	...	...	...	...	...
65	67.9	3.0	0.0	1.05	42.03	12.7
66	42.5	16.0	0.0	1.21	42.82	9.8
67	40.1	12.0	0.0	1.12	36.93	11.7
69	43.7	14.0	0.0	0.59	38.48	14.3
70	47.9	18.0	0.0	1.01	39.25	11.2

[5697 rows x 12 columns]

Those results will be stored in a excel file if anybody needs it in the future

	L1	Année	Allure	Vit. 20m	Vit. 50m	elopperCo	centeVertic	Souplesse	Equilibre	oordinatio
2	M	2019	21	3,39	6,59	62	43	-19	5	28,6
3	M	2019	14,5	3,98	7,32	34	37,1	6	4	30,1
4	M	2019	18,5	3,53	6,69	58	41	-2	10	29,8
5	M	2019	16	4,32	7,9	44	24,3	-32	10	29,1
6	M	2019	17	3,95	7,2	75,5	36,1	1	10	29,4
8	M	2019	17	3,47	6,53	50	47,1	-15	10	25,9
10	M	2019	16	3,76	6,86	76	40	-7	7	28,8
11	M	2019	13	3,72	6,93	67	43	-16	7	25,8
12	M	2019	17	3,91	7,22	82	39,8	0	5	31,8
13	M	2019	17,5	3,65	6,77	46	44,4	-13	3	24,6
15	M	2019	19	3,41	6,47	105	56	-5,5	6	24,8
16	F	2019	14,5	4,17	7,71	39,5	34,8	-15	8	29,5
17	F	2019	14,5	4,75	8,62	26,5	28,2	2	10	30,1
20	M	2019	19	3,26	6,42	70	53,9	-13	6	27,7
22	F	2019	16	3,76	6,99	46	42,8	3	0	24,8
24	M	2019	16,5	3,87	7,09	67	36,5	-4	10	27
25	M	2019	18,5	3,54	6,72	48	46,1	8,5	10	28,4
26	M	2019	19,5	3,72	6,73	55	36,5	-3	10	26,4
34	F	2019	15,5	3,89	7,17	29	33,5	-1	2	29,1
35	F	2019	13,5	3,59	8,03	40	27,7	-12	8	29,6
36	M	2019	15,5	3,08	6,74	55	35,9	-5	5	25,9
37	M	2019	19,5	3,22	6,83	44	46,1	-5	7	26,7
38	M	2019	15,5	3,39	7,42	53	36	-17	10	27,8
39	F	2019	17	3,41	7,18	51	43,4	7	0	25,8
40	F	2019	13,5	3,43	7,5	35	33,9	-1	4	24,9
42	M	2019	19	3,23	6,89	58	36,9	-6	6	26
44	M	2019	20,5	2,83	5,99	73	55,2	-11	3	23,2
45	F	2019	16,5	3,41	7,14	41	45	5	0	27,3

II) Primary˙results

1.Statistics˙on˙every˙elements:

Now that we have a clean data frame with no issue in it we can obtain some basics stats on it, the mean, the max the min, the standard deviation, the quartile. For every columns of data and either for the whole data or for any years specifically.

So first there is the stats for the whole data frame.

	Vit. 20m	Vit. 50m	Développer\nCouché	Détente\nVerticale	\
count	5660.000000	5660.000000	5664.000000	5661.000000	
mean	21.421191	25.157714	52.397952	45.604769	
std	1.908096	2.234903	18.562243	10.871623	
min	12.880143	16.901408	0.000000	0.000000	
25%	20.055710	23.468057	38.000000	38.000000	
50%	21.752266	25.677600	50.000000	45.000000	
75%	22.857143	26.835643	62.625000	53.000000	
max	30.000000	30.821918	135.000000	88.000000	

	Souplesse	Equilibre	Coordination	50M	BCPE\n
count	5661.000000	5662.000000	5659.000000	5192.000000	5343.000000
mean	11.436513	4.365772	18.974299	4.372750	9.877119
std	11.634584	2.918171	5.499300	0.827566	1.824550
min	-42.000000	0.000000	0.000000	0.900000	2.400000
25%	5.000000	2.000000	15.595000	3.811742	8.666667
50%	14.000000	4.000000	16.590000	4.376368	9.930000
75%	20.000000	6.000000	21.420000	4.927457	11.200000
max	37.000000	14.000000	43.370000	7.478189	15.000000

Here i took the example of the year 2005

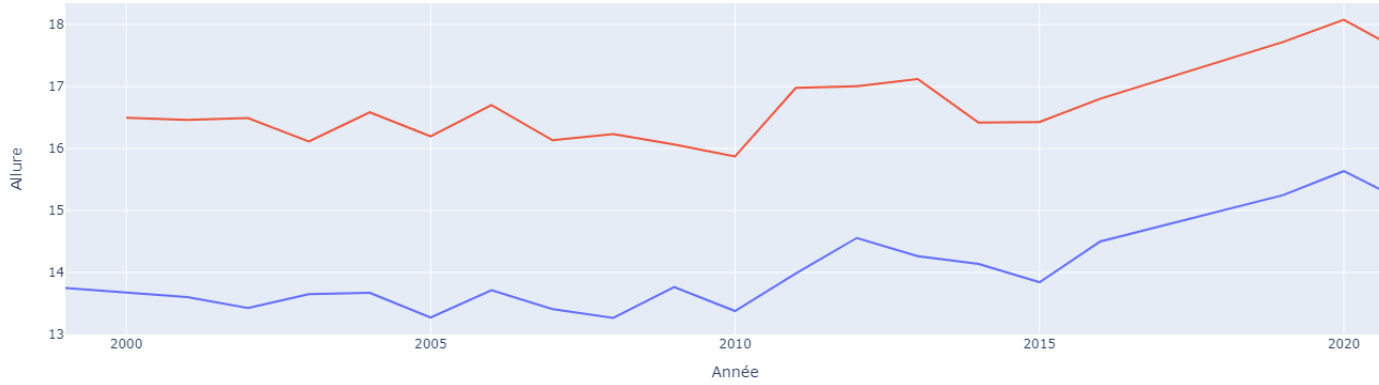
	Vit. 20m	Vit. 50m	Développer\nCouché	Détente\nVerticale	\
count	198.000000	198.000000	198.000000	198.000000	
mean	21.363247	24.726088	49.823232	53.601010	
std	1.563318	2.068530	15.588101	8.459962	
min	13.584906	19.911504	20.000000	35.000000	
25%	20.224719	22.944579	40.000000	48.000000	
50%	21.785224	25.316456	50.000000	53.500000	
75%	22.429907	26.239067	60.000000	59.000000	
max	24.489796	28.800000	110.000000	74.000000	

	Souplesse	Equilibre	Coordination	50M	BCPE\n
count	198.000000	198.000000	198.000000	198.000000	198.000000
mean	17.186869	4.818182	16.257273	4.356393	9.962290
std	6.482341	3.142516	0.926124	0.728816	1.607404
min	1.000000	0.000000	14.340000	2.844950	5.400000
25%	13.000000	2.000000	15.580000	3.778541	8.866667
50%	17.000000	5.000000	16.120000	4.343646	9.933333
75%	21.000000	7.000000	16.877500	4.867178	11.066667
max	35.000000	10.000000	18.930000	6.460876	14.200000

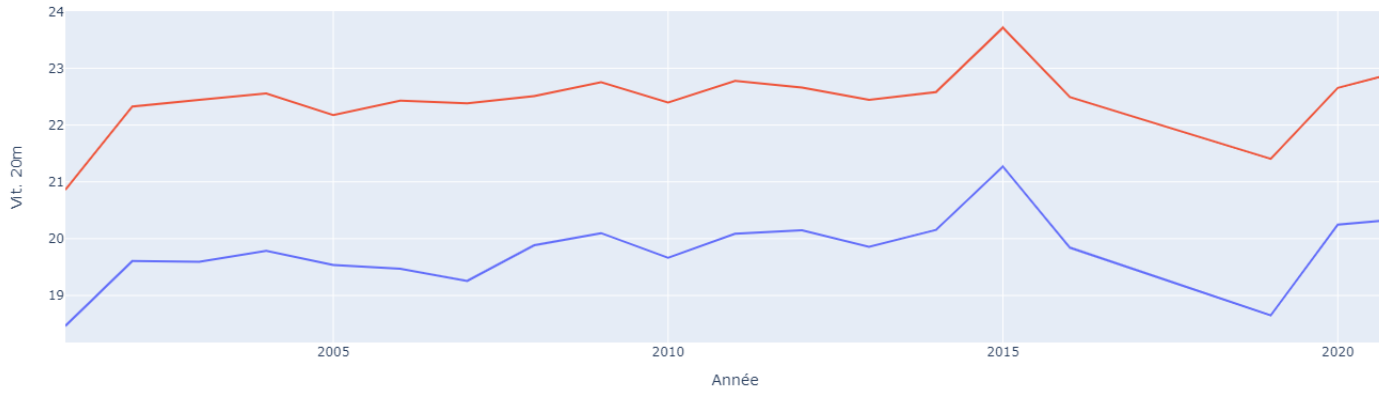
2.Graphs:

For each variable we choose now we can the see the evolution of the mean during the 20 years span, it will help to decide what are are our hypothesis for the more complex analysis yet to come.

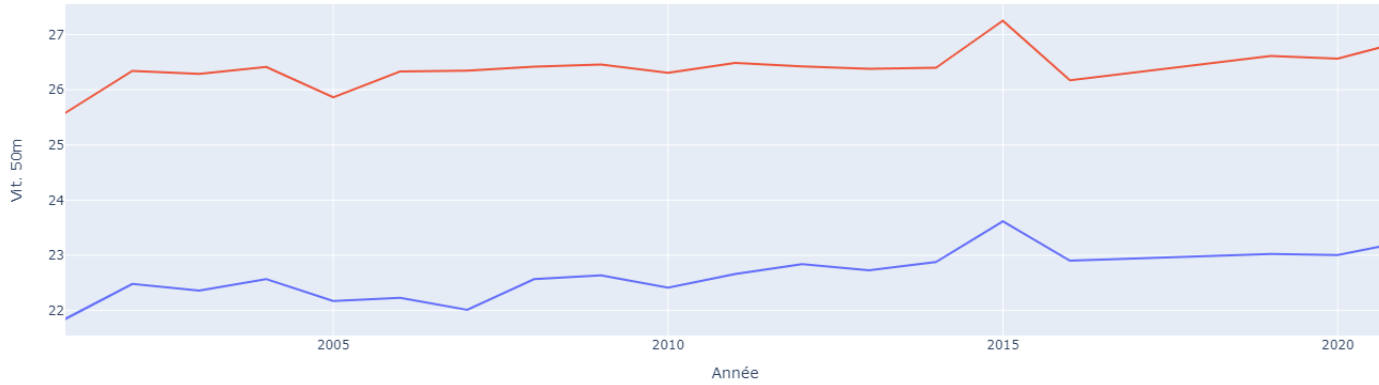
The endurance test with VmA measurements



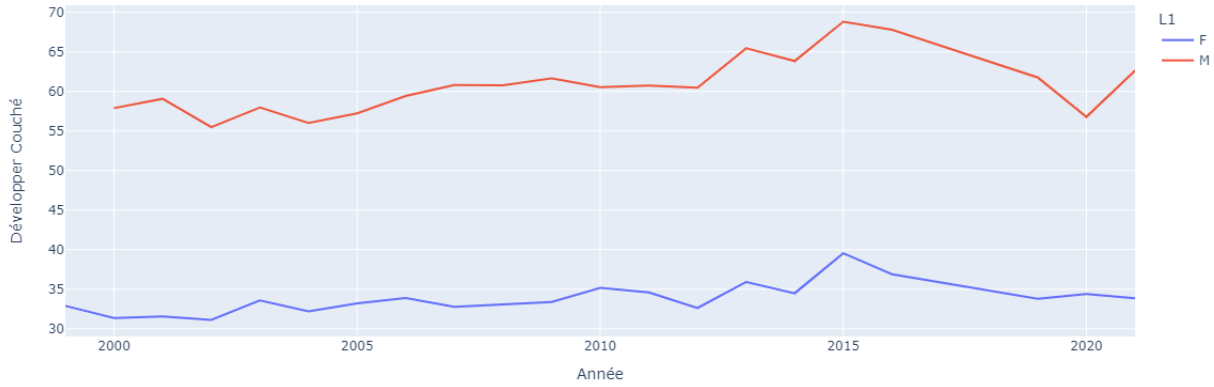
The speed on 20 meters (in km/h)



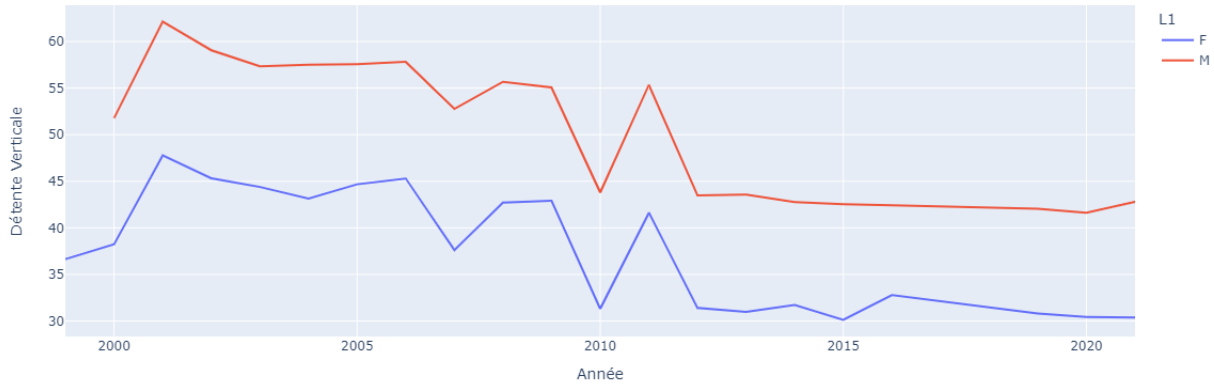
The speed on 50 meters (in km/h)



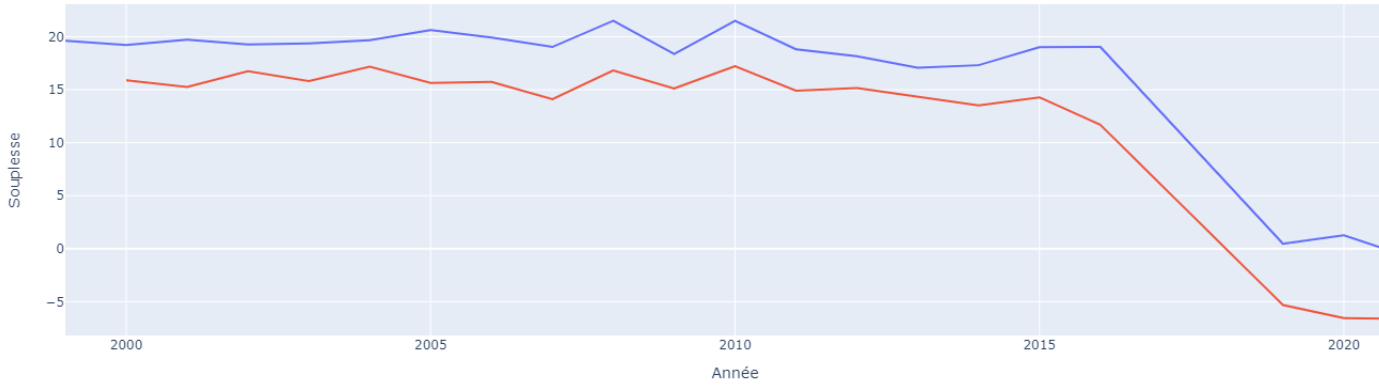
The bench press (in kg)



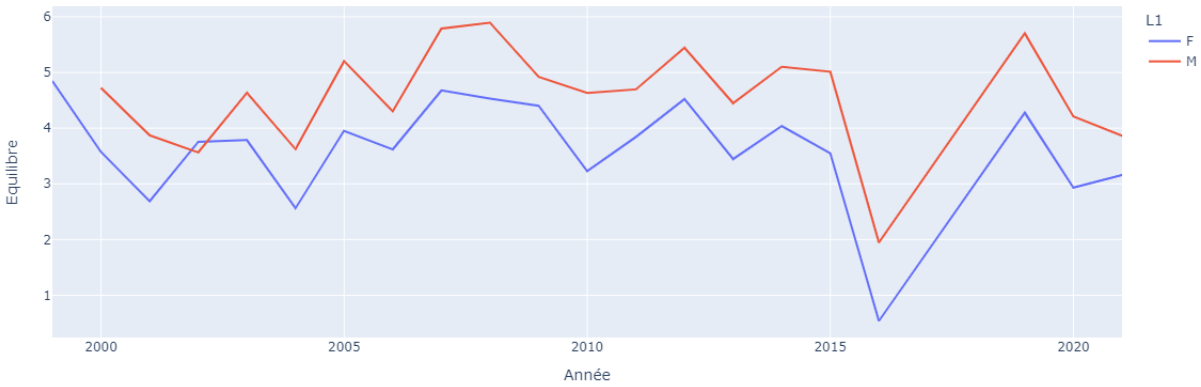
The vertical elasticity (in cm)



The flexibility (in cm)

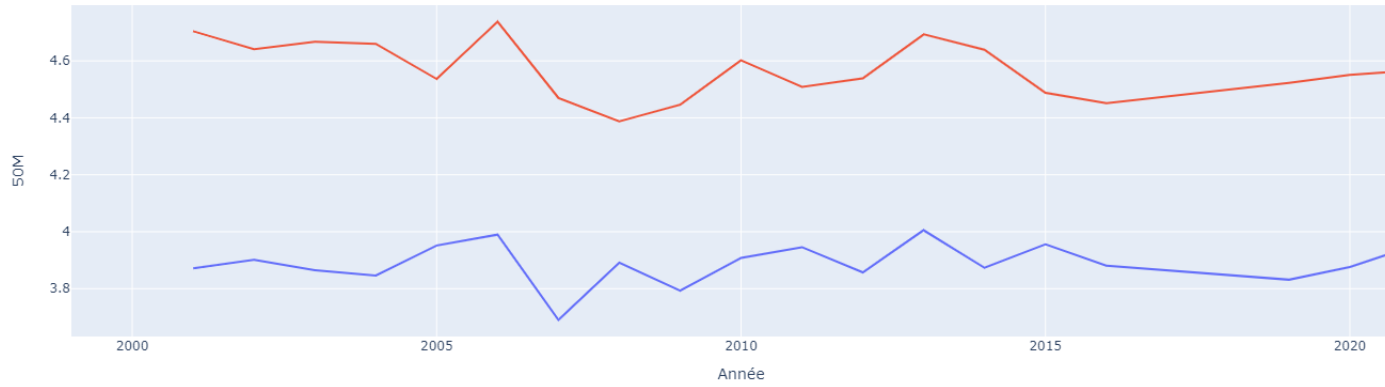


The balance (in number of fall)

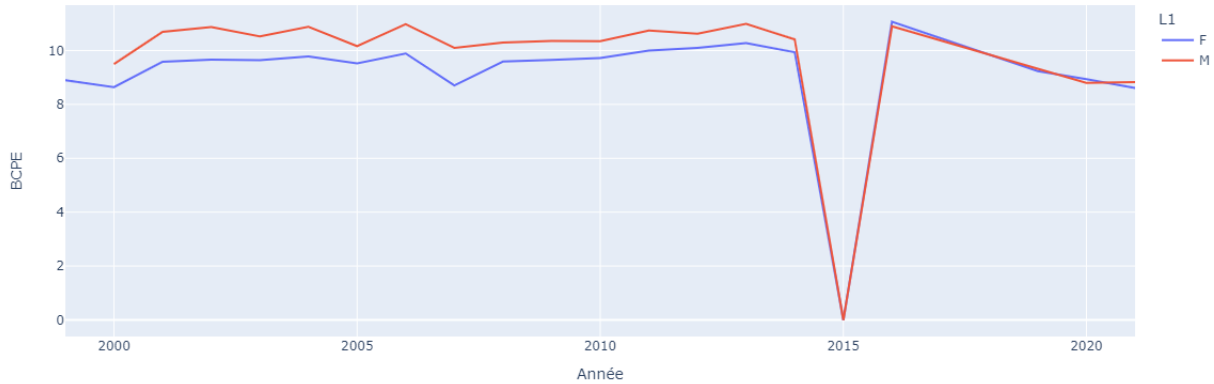


The coordination

The swimming speed on 50 meters (in km/h)



The overall mark on the exam (with the missing data of 2015)



III) Advanced results

Now that we saw the results of simple stats we can make greater assumptions about the data,depending on which element it is.

The Covid had an impact on the trials:
- Speed on 20 meters.
- The bench press for the men.
- The flexibility.
-The coordination (positive impact)
- The swimming speed on 50 meters.
- The mark of the exam.

The Covid had no impact on :
-The balance.
-Speed on 50 meters.
-Vertical elasticity
-The endurance test

Now we got our hypothesis for our tests and we try to conduct it on 2 of them : the Anova test and the Friedman Anova test. We want to verify the result that we can see on the means graph and the general idea that we suppose on it. To do so we use the Anova test and in case the Anova test does not conclude due to problems on the value of the data the Friedman anova test.

1.ANOVA:

To perform an ANOVA we have to check several properties first.
The fact that the different samples (here the result for each years) are independent, this is obviously the fact because results from on year can not influence one on an other year.
The equality of the variance (which should be the case because if the distribution of the performances) but we will verify it with a Bartlett test
The normality of the rest with a Shapiro test.
If these three condition are met the result of the ANOVA will be relevant for our study.
The expected result is to find that the ANOVA says that the means are not the same and then with a some t-tests we would like to see that the covid years are the one which prevent it.
Unfortunately after trying to do the Bartlett test it is shown that the data had way too much disparities in the variance and the test was not conclusive. Because of that the ANOVA test could not be used.

```
BartlettResult(statistic=105.35061387554663, pvalue=1.3725740624317382e-13)
BartlettResult(statistic=233.22310424096656, pvalue=2.7021497172996094e-38)
BartlettResult(statistic=31.0424649928976, pvalue=0.0546323388259929)
BartlettResult(statistic=76.23706686737647, pvalue=1.6925147089290268e-08)
BartlettResult(statistic=47.02583975351453, pvalue=0.0005817691000910358)
BartlettResult(statistic=58.309255277075465, pvalue=1.2966367336125631e-05)
BartlettResult(statistic=58.0598747449616, pvalue=1.4156466423357981e-05)
BartlettResult(statistic=1224.656921598954, pvalue=3.9686725976328 91e-247)
BartlettResult(statistic=nan, pvalue=nan)
BartlettResult(statistic=nan, pvalue=nan)
```

2.Friedman˙ANOVA:

So now we have to try the Friedmann Anova.

```python
df=pd.read_excel('CleanData.xlsx')
for j in liste2:
    print(pg.friedman(data=df, dv=j, within="Année", subject='L1'))
```

✓ 0.9s

```
          Source    W  ddof1     Q    p-unc
Friedman   Année  1.0     20  20.0  0.45793
          Source    W  ddof1     Q    p-unc
Friedman   Année  1.0     20  20.0  0.45793
          Source    W  ddof1     Q    p-unc
Friedman   Année  1.0     20  20.0  0.45793
          Source    W  ddof1     Q    p-unc
Friedman   Année  1.0     20  20.0  0.45793
          Source    W  ddof1     Q    p-unc
Friedman   Année  1.0     20  20.0  0.45793
          Source    W  ddof1     Q    p-unc
Friedman   Année  1.0     20  20.0  0.45793
          Source    W  ddof1     Q    p-unc
Friedman   Année  1.0     20  20.0  0.45793
          Source    W  ddof1     Q    p-unc
Friedman   Année  1.0     20  20.0  0.45793
          Source         W  ddof1          Q     p-unc
Friedman   Année  0.624561     18  22.484211  0.211195
          Source    W  ddof1     Q     p-unc
Friedman   Année  1.0     19  19.0  0.456836
```

As seen on the resulst we can say with certitude that years have a huge impact on the different results, but now we have to see if the years of covid are the one who are impacting the most the results.

```
    for j in liste2:
        print(sp.posthoc_conover_friedman(a=df, y_col=j, group_col="Année", block_col="L1",
                                          p_adjust="fdr_bh", melted=True))
```

✓ 0.5s

Output exceeds the size limit. Open the full output data in a text editor

	2019	2020	2021	2016	2014	\
2019	1.000000e+00	2.035265e-01	9.897003e-03	8.331350e-11	8.075968e-09	
2020	2.035265e-01	1.000000e+00	3.999037e-04	2.919839e-11	1.203781e-09	
2021	9.897003e-03	3.999037e-04	1.000000e+00	1.203781e-09	1.614117e-06	
2016	8.331350e-11	2.919839e-11	1.203781e-09	1.000000e+00	3.796267e-04	
2014	8.075968e-09	1.203781e-09	1.614117e-06	3.796267e-04	1.000000e+00	
2013	3.513523e-07	2.887168e-08	2.124163e-04	2.682888e-06	4.633186e-02	
2012	2.157445e-08	2.494735e-09	5.907866e-06	9.125431e-05	5.604948e-01	
2011	1.184382e-08	1.641159e-09	2.652568e-06	2.165500e-04	8.297358e-01	
2010	2.183323e-09	4.997713e-10	2.678549e-07	3.052303e-03	4.085819e-01	
2009	8.034542e-10	2.083853e-10	5.042630e-08	2.236901e-02	1.038349e-01	
2008	4.997713e-10	1.380455e-10	2.178614e-08	6.161582e-02	3.889760e-02	
2007	4.781827e-10	1.233915e-10	1.552442e-08	9.226501e-02	2.505166e-02	
2006	1.776153e-09	4.781827e-10	2.012690e-07	4.435795e-03	3.335907e-01	
2005	5.420242e-10	1.567180e-10	3.014483e-08	4.036567e-02	5.993253e-02	
2004	2.345867e-09	5.110772e-10	3.011094e-07	2.626834e-03	4.456467e-01	
2003	2.494735e-09	5.367468e-10	3.484271e-07	2.212422e-03	4.912190e-01	
2002	1.184382e-08	1.641159e-09	2.610959e-06	2.220293e-04	8.394676e-01	
2001	6.071523e-09	1.001509e-09	1.107623e-06	5.708762e-04	8.647102e-01	
2000	1.599841e-07	1.436254e-08	7.686269e-05	6.915101e-06	1.081566e-01	
1999	8.236018e-11	2.919839e-11	9.463450e-10	8.413683e-01	2.226395e-04	
2015	3.789352e-08	4.412770e-09	1.311527e-05	3.990350e-05	3.596460e-01	

We can see here that most of the variation are between the group of year (2020,2021) and the rest which means the results have indeed been impacted by the covid. We have to check this matrix for every trials and then we will be able to conclude on the impact of covid.

Now we can try again with the gender to know if the covid had a different impact on the results for every gender.

IV) Conclusion

In conclusion we can see that the covid had an impact on the students and their physical performances and it had different impact on the women and the men. But because the data that I got were not perfect for example missing two years, or some the results were not exploitable, having to measure exactly how it did can be a bit tricky, since the ANOVA did not work, and the Friedmann can be quite messy to interpret there might be others test which could give more information on the exact impact of the covid.

V) Bibliography

https://www.python-simple.com/python/statsmodels

https://stackoverflow.com

https://fr.wikipedia.org/wiki/ANOVA/de/Friedman

https://fr.wikipedia.org/wiki/Analyse/de/la/variance

https://www.reneshbedre.com/blog/friedman-test-python.html