```
In [1]: import pandas as pd
```

```
In [13]: df = pd.read_csv("realdata0.csv")
```

```
In [14]: for i in range(20):
             print(i)
             df = df.merge(pd.read_csv("realdata"+ str(i)+".csv", low_memory=False), how="outer")
```

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
```

```
In [20]: df
```

Out[20]:

	Unnamed: 0	index	term	course	Course	Instructor	Qtr/Yr	Enrollment	CAPEs Returned	Question	...	6	7
0	0	0.0	WI22	AAS 10	AAS 10 (A01) - Intro/African-American Studies	Butler, Elizabeth Annette	WI22	66.0	38.0	General Info	...	NaN	NaN
1	1	0.0	WI22	AAS 10	NaN	NaN	NaN	NaN	NaN	Your class level is	...	NaN	NaN
2	2	0.0	WI22	AAS 10	NaN	NaN	NaN	NaN	NaN	Your reason for taking this class is	...	NaN	NaN
3	3	0.0	WI22	AAS 10	NaN	NaN	NaN	NaN	NaN	What grade do you expect in this class?	...	NaN	NaN
4	4	0.0	WI22	AAS 10	NaN	NaN	NaN	NaN	NaN	I learned a great deal from this course.	...	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1034521	100	NaN	FA17	LIHL 132F	NaN	NaN	NaN	NaN	NaN	Grade received	...	NaN	NaN
1034522	101	NaN	FA17	LIHL 132F	NaN	NaN	NaN	NaN	NaN	Grade received	...	NaN	NaN
1034523	102	NaN	FA17	LIHL 132F	NaN	NaN	NaN	NaN	NaN	Grade received	...	NaN	NaN
1034524	103	NaN	FA17	LIHL 132F	NaN	NaN	NaN	NaN	NaN	Grade received	...	Custom Question 5	Custom Question 5
1034525	104	NaN	FA17	LIHL 132F	NaN	NaN	NaN	NaN	NaN	Grade received	...	NaN	NaN

1034526 rows × 67 columns

Cleaning

Get rid everything that is Fa17, since it is only paritally collected

```
In [19]: data = df[df["term"] != "FA17"]
         data
```

Out[19]:

	Unnamed: 0	index	term	course	Course	Instructor	Qtr/Yr	Enrollment	CAPEs Returned	Question	...	6	7	8
0	0	0.0	WI22	AAS 10	AAS 10 (A01) - Intro/African-American Studies	Butler, Elizabeth Annette	WI22	66.0	38.0	General Info	...	NaN	NaN	NaN
1	1	0.0	WI22	AAS 10	NaN	NaN	NaN	NaN	NaN	Your class level is	...	NaN	NaN	NaN
2	2	0.0	WI22	AAS 10	NaN	NaN	NaN	NaN	NaN	Your reason for taking this class is	...	NaN	NaN	NaN
3	3	0.0	WI22	AAS 10	NaN	NaN	NaN	NaN	NaN	What grade do you expect in this class?	...	NaN	NaN	NaN
4	4	0.0	WI22	AAS 10	NaN	NaN	NaN	NaN	NaN	I learned a great deal from this course.	...	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1006563	23	0.0	WI18	WCWP 10B	NaN	NaN	NaN	NaN	NaN	Instructor is effective in promoting academic ...	...	NaN	NaN	NaN
1006564	24	0.0	WI18	WCWP 10B	NaN	NaN	NaN	NaN	NaN	The instructor practiced effective teaching st...	...	NaN	NaN	NaN
1006565	25	0.0	WI18	WCWP 10B	NaN	NaN	NaN	NaN	NaN	Do you recommend this professor overall?	...	NaN	NaN	NaN
1006566	0	NaN	WI18	WCWP 10B	NaN	NaN	NaN	NaN	NaN	Grade received	...	NaN	NaN	NaN
1006567	1	NaN	WI18	WCWP 10B	NaN	NaN	NaN	NaN	NaN	Grade received	...	NaN	NaN	NaN

1006568 rows × 67 columns

In this analysis, we are just focusing on real scores, thus, we would just analyze `Grade received`.

```
In [24]: data = data[data.Question == "Grade received"]
         data
```

Out[24]:

	Unnamed: 0	index	term	course	Course	Instructor	Qtr/Yr	Enrollment	CAPEs Returned	Question	...	6	7
26	0	NaN	WI22	AAS 10	NaN	NaN	NaN	NaN	NaN	Grade received	...	NaN	NaN
27	1	NaN	WI22	AAS 10	NaN	NaN	NaN	NaN	NaN	Grade received	...	NaN	NaN
54	0	NaN	WI22	ANAR 116	NaN	NaN	NaN	NaN	NaN	Grade received	...	NaN	NaN
55	1	NaN	WI22	ANAR 116	NaN	NaN	NaN	NaN	NaN	Grade received	...	NaN	NaN
56	2	NaN	WI22	ANAR 116	NaN	NaN	NaN	NaN	NaN	Grade received	...	PLEASE COMMENT ON THE FOLLOWING:	PLEASE COMMENT ON THE FOLLOWING:
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1006434	1	NaN	WI18	WCWP 100	NaN	NaN	NaN	NaN	NaN	Grade received	...	NaN	NaN
1006538	0	NaN	WI18	WCWP 10A	NaN	NaN	NaN	NaN	NaN	Grade received	...	NaN	NaN
1006539	1	NaN	WI18	WCWP 10A	NaN	NaN	NaN	NaN	NaN	Grade received	...	NaN	NaN
1006566	0	NaN	WI18	WCWP 10B	NaN	NaN	NaN	NaN	NaN	Grade received	...	NaN	NaN
1006567	1	NaN	WI18	WCWP 10B	NaN	NaN	NaN	NaN	NaN	Grade received	...	NaN	NaN

495634 rows × 67 columns

```
In [31]: data.columns
```

```
Out[31]: Index(['Unnamed: 0', 'index', 'term', 'course', 'Course', 'Instructor',
       'Qtr/Yr', 'Enrollment', 'CAPEs Returned', 'Question', 'Freshman',
       'Sophomore', 'Junior', 'Senior', 'Graduate', 'Extension', 'Visitor',
       'Major', 'Minor', 'Gen. Ed.', 'Elective', 'Interest', 'A', 'B', 'C',
       'D', 'F', 'P', 'NP', 'Strongly Disagree', 'Disagree',
       'Neither Agree nor Disagree', 'Agree', 'Strongly Agree',
       'Not Applicable', '0-1', '2-3', '4-5', '6-7', '8-9', '10-11', '12-13',
       '14-15', '16-17', '18-19', '20 or more', 'Very Rarely',
       'Some of the Time', 'Most of the Time', 'Yes', 'No', '0', '1', '2', '3',
       '4', '5', '6', '7', '8', '9', '10', '11', '12', '13', '14', '15'],
      dtype='object')
```

Since we are only checking grades, we only need course, term, and grade A-F and P/NP

```
In [40]: data = data[["term", "course", 'A', 'B', 'C',
              'D', 'F', 'P', 'NP']]
         data
```

Out[40]:

	term	course	A	B	C	D	F	P	NP
26	WI22	AAS 10	53	2	1	2	1	6	1
27	WI22	AAS 10	80 %	3 %	2 %	3 %	2 %	9 %	2 %
54	WI22	ANAR 116	NaN	NaN	NaN	NaN	NaN	NaN	NaN
55	WI22	ANAR 116	NaN	NaN	NaN	NaN	NaN	NaN	NaN
56	WI22	ANAR 116	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...
1006434	WI18	WCWP 100	25 %	65 %	10 %	0 %	0 %	0 %	0 %
1006538	WI18	WCWP 10A	45	162	12	3	3	0	0
1006539	WI18	WCWP 10A	20 %	72 %	5 %	1 %	1 %	0 %	0 %
1006566	WI18	WCWP 10B	84	230	41	1	1	0	0
1006567	WI18	WCWP 10B	24 %	64 %	11 %	0 %	0 %	0 %	0 %

495634 rows × 9 columns

```
In [53]: data.isna().any()
```

```
Out[53]: term       False
course     False
A           True
B           True
C           True
D           True
F           True
P           True
NP          True
dtype: bool
```

Drop na here, since we are sure that the nan data only exist in the grade data, thus we could drop any na. Since these are corrupted.

```
In [56]: data = data.dropna()
         data
```

Out[56]:

	term	course	A	B	C	D	F	P	NP
26	WI22	AAS 10	53	2	1	2	1	6	1
27	WI22	AAS 10	80 %	3 %	2 %	3 %	2 %	9 %	2 %
316	WI22	ANAR 135	22	0	0	0	0	0	0
317	WI22	ANAR 135	100 %	0 %	0 %	0 %	0 %	0 %	0 %
344	WI22	ANAR 164	23	11	1	1	3	0	0
...	...	...	...	...	...	...	...	...	...
1006434	WI18	WCWP 100	25 %	65 %	10 %	0 %	0 %	0 %	0 %
1006538	WI18	WCWP 10A	45	162	12	3	3	0	0
1006539	WI18	WCWP 10A	20 %	72 %	5 %	1 %	1 %	0 %	0 %
1006566	WI18	WCWP 10B	84	230	41	1	1	0	0
1006567	WI18	WCWP 10B	24 %	64 %	11 %	0 %	0 %	0 %	0 %

30830 rows × 9 columns

Using the % value is slightly easier, so we would only need the % values.

```
In [60]: data = data[data.A.str.contains("%")]
         data
```

Out[60]:

	term	course	A	B	C	D	F	P	NP
27	WI22	AAS 10	80 %	3 %	2 %	3 %	2 %	9 %	2 %
317	WI22	ANAR 135	100 %	0 %	0 %	0 %	0 %	0 %	0 %
345	WI22	ANAR 164	59 %	28 %	3 %	3 %	8 %	0 %	0 %
504	WI22	ANBI 116	85 %	10 %	2 %	0 %	1 %	2 %	0 %
663	WI22	ANBI 131	55 %	19 %	9 %	2 %	11 %	2 %	1 %
...	...	...	...	...	...	...	...	...	...
1006432	WI18	WCWP 100	25 %	65 %	10 %	0 %	0 %	0 %	0 %
1006433	WI18	WCWP 100	25 %	65 %	10 %	0 %	0 %	0 %	0 %
1006434	WI18	WCWP 100	25 %	65 %	10 %	0 %	0 %	0 %	0 %
1006539	WI18	WCWP 10A	20 %	72 %	5 %	1 %	1 %	0 %	0 %
1006567	WI18	WCWP 10B	24 %	64 %	11 %	0 %	0 %	0 %	0 %

15418 rows × 9 columns

Let's strip the % so that it would be a int

```
In [65]: data.columns
```

```
Out[65]: Index(['term', 'course', 'A', 'B', 'C', 'D', 'F', 'P', 'NP'], dtype='object')
```

```
In [66]: for column in data.columns[2:]:
             data[column] = data[column].str.replace("%", "").astype("int")
```

```
In [67]: data
```

Out[67]:

	term	course	A	B	C	D	F	P	NP
27	WI22	AAS 10	80	3	2	3	2	9	2
317	WI22	ANAR 135	100	0	0	0	0	0	0
345	WI22	ANAR 164	59	28	3	3	8	0	0
504	WI22	ANBI 116	85	10	2	0	1	2	0
663	WI22	ANBI 131	55	19	9	2	11	2	1
...	...	...	...	...	...	...	...	...	...
1006432	WI18	WCWP 100	25	65	10	0	0	0	0
1006433	WI18	WCWP 100	25	65	10	0	0	0	0
1006434	WI18	WCWP 100	25	65	10	0	0	0	0
1006539	WI18	WCWP 10A	20	72	5	1	1	0	0
1006567	WI18	WCWP 10B	24	64	11	0	0	0	0

15418 rows × 9 columns

Let's drop summer quarters, because it is not that useful

```
In [80]: data = data[data["term"].str.contains("SP|WI|FA")]
         data
```

Out[80]:

	term	course	A	B	C	D	F	P	NP	real	pnp
27	WI22	AAS 10	80	3	2	3	2	9	2	90	11
317	WI22	ANAR 135	100	0	0	0	0	0	0	100	0
345	WI22	ANAR 164	59	28	3	3	8	0	0	101	0
504	WI22	ANBI 116	85	10	2	0	1	2	0	98	2
663	WI22	ANBI 131	55	19	9	2	11	2	1	96	3
...	...	...	...	...	...	...	...	...	...	...	...
1006432	WI18	WCWP 100	25	65	10	0	0	0	0	100	0
1006433	WI18	WCWP 100	25	65	10	0	0	0	0	100	0
1006434	WI18	WCWP 100	25	65	10	0	0	0	0	100	0
1006539	WI18	WCWP 10A	20	72	5	1	1	0	0	99	0
1006567	WI18	WCWP 10B	24	64	11	0	0	0	0	99	0

13435 rows × 11 columns

EDA

Let's graph how many people receive real grades, and np between the terms

```
In [81]: data["real"] = data.A + data.B + data.C + data.D + data.F
         data["pnp"] = data.P + data.NP
```

```
In [82]: data
```

Out[82]:

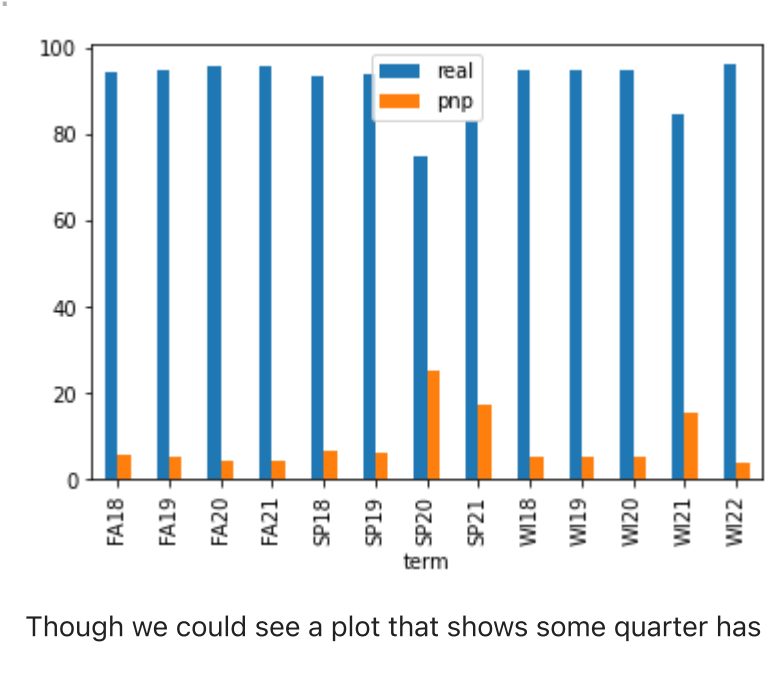
	term	course	A	B	C	D	F	P	NP	real	pnp
27	WI22	AAS 10	80	3	2	3	2	9	2	90	11
317	WI22	ANAR 135	100	0	0	0	0	0	0	100	0
345	WI22	ANAR 164	59	28	3	3	8	0	0	101	0
504	WI22	ANBI 116	85	10	2	0	1	2	0	98	2
663	WI22	ANBI 131	55	19	9	2	11	2	1	96	3
...	...	...	...	...	...	...	...	...	...	...	...
1006432	WI18	WCWP 100	25	65	10	0	0	0	0	100	0
1006433	WI18	WCWP 100	25	65	10	0	0	0	0	100	0
1006434	WI18	WCWP 100	25	65	10	0	0	0	0	100	0
1006539	WI18	WCWP 10A	20	72	5	1	1	0	0	99	0
1006567	WI18	WCWP 10B	24	64	11	0	0	0	0	99	0

13435 rows × 11 columns

Let's use groupby term and see what term has the most people pnp

```
In [83]: data.groupby("term").mean()[["real", "pnp"]].plot(kind="bar")
```

```
Out[83]: <AxesSubplot:xlabel='term'>
```



Though we could see a plot that shows some quarter has some spikes, we wanted to sort it in a better way to present it.

Sorting

```
In [84]: qtr_lst = []
         for i in range(18, 23):
             qtr_lst.append("WI"+str(i))
             qtr_lst.append("SP"+str(i))
             qtr_lst.append("FA"+str(i))
```

```
In [87]: qtr_lst
```

```
Out[87]: ['WI18',
 'SP18',
 'FA18',
 'WI19',
 'SP19',
 'FA19',
 'WI20',
 'SP20',
 'FA20',
 'WI21',
 'SP21',
 'FA21',
 'WI22',
 'SP22',
 'FA22']
```

```
In [88]: qtr_lst.pop()
         qtr_lst.pop()
```

```
Out[88]: 'SP22'
```

```
In [89]: qtr_lst
```

```
Out[89]: ['WI18',
 'SP18',
 'FA18',
 'WI19',
 'SP19',
 'FA19',
 'WI20',
 'SP20',
 'FA20',
 'WI21',
 'SP21',
 'FA21',
 'WI22']
```

```
In [96]: qtr_df = pd.DataFrame(qtr_lst)
         qtr_df = qtr_df.reset_index().set_index(0)
         qtr_df
```

Out[96]:

0	index
WI18	0
SP18	1
FA18	2
WI19	3
SP19	4
FA19	5
WI20	6
SP20	7
FA20	8
WI21	9
SP21	10
FA21	11
WI22	12

```
In [100…: groupbydf = data.groupby("term").mean()[["real", "pnp"]].reset_index()
          groupbydf
```

Out[100…:

	term	real	pnp
0	FA18	94.316667	5.701042
1	FA19	94.634597	5.362488
2	FA20	95.730101	4.248856
3	FA21	95.500000	4.516605
4	SP18	93.293989	6.689617
5	SP19	93.731633	6.281633
6	SP20	74.894373	25.102665
7	SP21	82.949758	17.074396
8	WI18	94.579894	5.450794
9	WI19	94.853147	5.146853
10	WI20	94.609152	5.381316
11	WI21	84.688263	15.308920
12	WI22	96.016588	3.983412

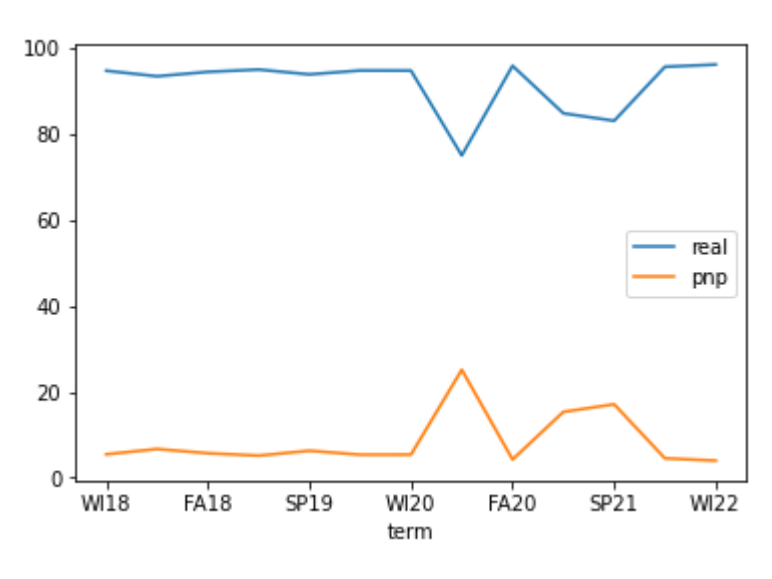
```
In [102…: groupbydf['term_num'] = groupbydf['term'].map(qtr_df['index'])
          groupbydf
```

Out[102…:

	term	real	pnp	term_num
0	FA18	94.316667	5.701042	2
1	FA19	94.634597	5.362488	5
2	FA20	95.730101	4.248856	8
3	FA21	95.500000	4.516605	11
4	SP18	93.293989	6.689617	1
5	SP19	93.731633	6.281633	4
6	SP20	74.894373	25.102665	7
7	SP21	82.949758	17.074396	10
8	WI18	94.579894	5.450794	0
9	WI19	94.853147	5.146853	3
10	WI20	94.609152	5.381316	6
11	WI21	84.688263	15.308920	9
12	WI22	96.016588	3.983412	12

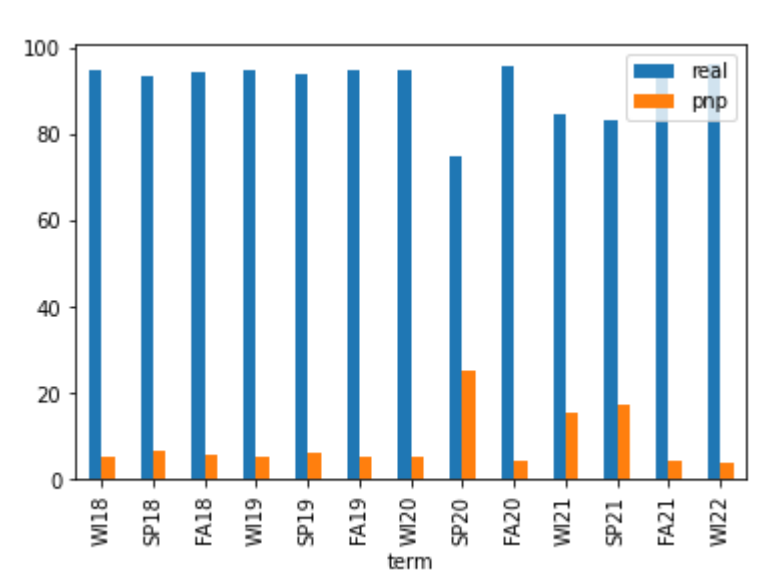
```
In [108…: groupbydf.sort_values(by="term_num").set_index("term")[["real", "pnp"]].plot(kind = "line")
```

```
Out[108…: <AxesSubplot:xlabel='term'>
```




```
In [110…: groupbydf.sort_values(by="term_num").set_index("term")[["real", "pnp"]].plot(kind = "bar")
```

```
Out[110…: <AxesSubplot:xlabel='term'>
```




```
In [ ]:
```