Universidade Federal do Pampa - Campus Alegrete

Graduação em Ciência da Computação

Graduação em Engenharia de Software

Mathias Baldissera, Wesley Ferreira

Análise de dados sobre evasão de discentes na Universidade Federal do Pampa

Alegrete, Rio Grande do Sul, Brasil

2 de Junho de 2018

Mathias Baldissera, Wesley Ferreira

Análise de dados sobre evasão de discentes na Universidade Federal do Pampa

Relatório final de pesquisa, apresentado como requisito parcial de aprovação na disciplina de Introdução a Análise de Dados

Universidade Federal do Pampa - Campus Alegrete

Graduação em Ciência da Computação

Graduação em Engenharia de Software

Alegrete, Rio Grande do Sul, Brasil

2 de Junho de 2018

Resumo

Devido aos custos e os investimentos de uma universidade pública, o grande número de evasões de discentes torna-se um tema de extrema relevância, já que parte desses investimentos acabam perdidos, sendo então necessário uma análise buscando o motivo dessas evasões e as estratégias para amenizar esse fenomeno. Esta pesquisa objetivou apresentar uma análise estatística deste fenômeno nos cursos de graduação da Universidade Federal do Pampa(Unipampa), dado que é perceptível em classe essa tendência. Para essa análise foi utilizada a linguagem de programação Python, dentre outras ferramentas, que estendem as funcionalidades da linguagem e que foram de essencial importância para a análise dos dados, como Pandas, Numpy e Matplotlib. Dentre algumas das conclusões nota-se que as cidades de Bagé e Alegrete tem, somando todos os seus cursos, os maiores índices de evasão, esta afirmação é corroborada também pela análise individual de cada curso, onde fica evidente que as áreas com um alto grau de matemática no currículo, um maior índice de evasão.

Palavras-chave: Evasão, Abandono, Cursos, Unipampa.

Sumário

1 Introdução

Devido os custos que as universidades públicas tem ao governo e dos investimentos que estas fazem em seus discentes, o grande número de evasões torna-se um tema de extrema relevância, já que parte desse dinheiro investido vem sendo perdido, sendo portanto necessário uma análise buscando o motivo dessas evasões e a elaboração de estratégia e estimulo para permanencia na universidade.

Nosso trabalho insere-se no contexto de fazer uma análise estatística dos dados de alunos matriculados e evadidos na Universidade Federal do Pampa(Unipampa) fornecidos pela mesma, na tentativa de identificação de padrões relacionais entre estes dados, diferentemente do viés adotado por José e Andreoli (2011) que faz uma análise de perfil de docentes, discentes e dirigentes.

Dentre os dados analisados por nós estão:

a) Planilhas com o Relatório Geral de Evasão dos períodos letivos compreendidos entre 1º semestre de 2006 a 1º semestre de 2011 de José e Andreoli (2011);

b) As planilha de todos os alunos regularmente matriculados de 2016 a 2018 de Unipampa (2018).

Um, intervalo, portanto, de 7 anos, com uma lacuna de 5 entre o primeiro semestre de 2011 e o segundo semestre de 2015, de dados não disponibilizado a priori publicamente.

2 Desenvolvimento

2.1 Análise inicial das bases de dados

Em uma análise inicial das bases de dados identificamos duas distribuições diferentes nos dados a serem avaliados, dentre estes 3 organizações diferentes de tabelas, a baixo um exemplo de cada um dos tipos diferentes:

a) Tabelas em pdf no artigo de José e Andreoli (2011) das paginas 30 à 35;

b) Tabelas em pdf no artigo de José e Andreoli (2011) das paginas 36 à 56;

c) Tabelas em formato .csv disponibilizados no sistema da universidade Unipampa (2018).

A base de dados no formato pdf analisada demonstra casos de evasão diretamente, no caso a) discriminado separadamente cada tipo de evasão em colunas diferentes, vide Figura 1, no caso b) os tipos de evasão são discriminados todos na mesma coluna, vide Figura 2, e a base de dados no formato csv, do caso c), analisada demonstra o total de alunos matriculados em cada um dos campus em determinado mês do ano, vide Figura 3.

Figura 1: Tabela de evasão do primeiro semestre 2006.

Evasão – 1º semestre de 2006

NOME_CURSO	Nº de alunos	Ocorrência/forma de evasão	Nº por Campus	Campus
		Classificado e Não Matriculado		
CURSO DE CIENCIA DA COMPUTACAO	1	1	7	Alegrete
CURSO DE ENGENHARIA CIVIL	4	4		
CURSO DE ENGENHARIA ELETRICA	2	2		
CURSO DE AGRONOMIA	1	1	1	Itaqui
CURSO DE JORNALISMO	2	2	5	São Borja
CURSO DE PUBLICIDADE E PROPAGANDA	2	2		
CURSO DE SERVICO SOCIAL	1	1		
CURSO DE CIENCIAS BIOLOGICAS - NUCLEO COMUM	5	5	10	São Gabriel
CURSO DE ENGENHARIA FLORESTAL	1	1		
CURSO DE GESTAO AMBIENTAL	4	4		
CURSO DE ENFERMAGEM	1	1	8	Uruguaiana
CURSO DE FARMACIA	3	3		
CURSO DE FISIOTERAPIA	4	4		
TOTAL	31			

Fonte: José e Andreoli (2011)

Figura 2: Tabela de evasão do primeiro semestre 2011.

Evasão – 1º semestre de 2011

Dados de evasão do período - 1º semestre de 2011				
NOME_CURSO	Total de estudantes	Ocorrência/forma de evasão	Nº por Campus	Campus
CURSO DE CIENCIA DA COMPUTACAO	6	Cancelamento=6	29	Alegrete
CURSO DE ENGENHARIA AGRÍCOLA	3	Abandono=2		
		Cancelamento=1		
CURSO DE ENGENHARIA CIVIL	5	Cancelamento=3		
		Transferência=2		
CURSO DE ENGENHARIA ELETRICA	8	Cancelamento=8		
CURSO DE ENGENHARIA MECANICA	5	Cancelamento=5		
CURSO DE ENGENHARIA DE SOFTWARE	2	Cancelamento=2		
	1	Transferência=1	14	Bagé
CURSO DE ENGENHARIA DE COMPUTACAO	2	Abandono=1		
		Cancelamento=1		
CURSO DE ENGENHARIA DE PRODUCAO	1	Classificado e Não Matriculado=1		
CURSO DE ENGENHARIA QUIMICA	2	Cancelamento=1		
		Transferência=1		
LICENCIATURA EM LETRAS-LÍNGUA PORTUGUESA E RESPECTIVAS LITERATURAS	1	Transferência=1		
CURSO DE LICENCIATURA EM FISICA	3	Cancelamento=3		
CURSO DE LICENCIATURA EM LETRAS - PORTUGUES E ESPANHOL	1	Cancelamento=1		
CURSO DE LICENCIATURA EM QUIMICA	3	Cancelamento=3		
CURSO DE LICENCIATURA EM CIENCIAS EXATAS	3	Cancelamento=3	15	Caçapava do Sul
CURSO DE GEOLOGIA	1	Cancelamento=1		
CURSO DE GEOFISICA	9	Cancelamento=7		
		Transferência=2		
CURSO SUPERIOR DE TECNOLOGIA EM MINERACAO	2	Cancelamento=2		
CURSO DE ZOOTECNIA	1	Cancelamento=1	1	Dom Pedrito
REGIME ESPECIAL DE GRADUACAO	3	Abandono=1	3	Regime especial de Graduação
		Cancelamento=1		
		Sem Matrícula=1		
CURSO DE AGRONOMIA	1	Cancelamento=1	1	Itaqui
BACHARELADO INTERDISCIPLINAR EM CIÊNCIA E TECNOLOGIA-DIURNO	2	Cancelamento=2	2	
CURSO DE NUTRIÇÃO	2	Cancelamento=2	2	
CURSO DE LICENCIATURA EM HISTÓRIA	3	Cancelamento=3	3	Jaguarão
CURSO DE LICENCIATURA EM LETRAS - ESPANHOL - NOTURNO	2	Cancelamento=2	2	
CURSO DE CIENCIAS SOCIAIS - CIENCIA POLITICA	1	Cancelamento=1	1	São Borja
CURSO DE JORNALISMO	1	Cancelamento=1	1	
CURSO DE PUBLICIDADE E PROPAGANDA	1	Cancelamento=1	1	
CURSO DE RELAÇÕES PÚBLICAS	2	Cancelamento=2	2	
CURSO DE SERVIÇO SOCIAL	1	Cancelamento=1	1	
CURSO DE BIOTECNOLOGIA	5	Transferência=5	8	São Gabriel
CURSO DE ENGENHARIA FLORESTAL	3	Cancelamento=1		
		Transferência=2		
CURSO DE ADMINISTRACAO - DIURNO	3	Cancelamento=1	3	Santana do Livramento

Fonte: José e Andreoli (2011)

Figura 3: Tabela de evasão de janeiro de 2018.

campus	nível	nome do curso	modalidade	turno	alunos matriculados
Campus Alegrete	Graduação	Ciência da Computação	Bacharelado	Noturno	148
Campus Alegrete	Graduação	Engenharia Agrícola	Bacharelado	Integral	185
Campus Alegrete	Graduação	Engenharia Civil	Bacharelado	Integral	259
Campus Alegrete	Graduação	Engenharia de Software	Bacharelado	Noturno	146
Campus Alegrete	Graduação	Engenharia de Telecomunicações	Bacharelado	Integral	116
Campus Alegrete	Graduação	Engenharia Elétrica	Bacharelado	Integral	223
Campus Alegrete	Graduação	Engenharia Mecânica	Bacharelado	Integral	211
Campus Alegrete	Pós-Graduação	ESPECIALIZAÇÃO EM ENGENHARIA ECONÔMICA	Especialização	Integral	19
Campus Alegrete	Pós-Graduação	Mestrado em Engenharia Elétrica	Mestrado	Integral	27
Campus Alegrete	Pós-Graduação	Mestrado em Engenharias	Mestrado	Integral	31
Campus Bagé	Graduação	Engenharia de Alimentos	Bacharelado	Integral	127
Campus Bagé	Graduação	Engenharia de Computação	Bacharelado	Noturno	187
Campus Bagé	Graduação	Engenharia de Energia	Bacharelado	Integral	167
Campus Bagé	Graduação	Engenharia de Produção	Bacharelado	Noturno	220
Campus Bagé	Graduação	Engenharia Química	Bacharelado	Integral	243
Campus Bagé	Graduação	Física	Licenciatura Plena	Integral	85
Campus Bagé	Graduação	Letras - Habilitação Português e Literaturas de Língua Portuguesa	Licenciatura Plena	Noturno	9
Campus Bagé	Graduação	Letras - Habilitação Português/Espanhol e Respectivas Literaturas	Licenciatura Plena	Noturno	6
Campus Bagé	Graduação	Letras - Habilitação Português/Inglês e Respectivas Literaturas	Licenciatura Plena	Noturno	8
Campus Bagé	Graduação	Letras - Línguas Adicionais: Inglês Espanhol e Respectivas Literaturas	Licenciatura Plena	Integral	111
Campus Bagé	Graduação	Letras - Português e Literaturas da Língua Portuguesa	Licenciatura Plena	Noturno	145
Campus Bagé	Graduação	Matemática	Licenciatura Plena	Noturno	110
Campus Bagé	Graduação	Música	Licenciatura Plena	Integral	71
Campus Bagé	Graduação	Química	Licenciatura Plena	Integral	104
Campus Bagé	Pós-Graduação	Especialização em Modelagem Computacional em Ensino - Experimentação e Simul	Especialização	Noturno	12
Campus Bagé	Pós-Graduação	Mestrado em Computação Aplicada	Mestrado	Integral	13
Campus Bagé	Pós-Graduação	Mestrado em Ensino	Mestrado	Integral	25
Campus Bagé	Pós-Graduação	Mestrado Profissional em Ensino de Ciências	Mestrado	Integral	46

Fonte: Unipampa (2018)

2.2 Perguntas que tivemos que responder

Depois de uma breve análise das tabelas foram levantadas algumas perguntas a serem respondidas antes de qualquer alteração nas bases de dados, as perguntas foram geradas com base no achamos que tenha relevância que pôde ser retirado de informação do dataset. São elas:

a) Qual as cidades com o maior índice de evasões?

b) Qual a categoria de maior evasão?

c) Qual o semestre de maior evasões?

d) Qual o curso de maior evasão?

e) Qual o turno de curso com maior evasão?

f) Comparação entre número de cursos em comparação ao número de evasões.

g) O maior número de evasões ocorre no primeiro ou mo segundo semestre do ano?

2.3 Ferramental utilizado

A limpeza, análise e plotagem das tabelas foi feita na linguagem Python 3, no ambiente de desenvolvimento Jupyter Notebook. No entanto, por uma delas estar em formato PDF, outros meios foram necessários para a extração desses dados, são eles(nesta ordem): Ferramenta de "Dividir PDF"da smallpdf(https://smallpdf.com/pt/dividir-pdf) que foi usada para a remoção de elementos textuais do PDF(deixando apenas as paginas relativas as tabelas); Ferramenta de "Converter PDF em Excel"da smallpdf(https://smallpdf.com/pt/pdf-para-excel) que fez o trabalho de extração das tabelas e geração uma planilha Excel com elas; Excel: para a exportação de todas as tabelas para .csv para poderem ser analisadas juntamente com o outro dataset.

Passo a passo da extração das tabelas do pdf:

a) Enviar o documento PDF para a ferramenta "Dividir PDF"e marcar selecionar paginas;

b) Selecionar da pagina 30 à pagina 56(paginas onde estão as tabelas);

c) Enviar o PDF gerado pelo "Dividir PDF"para o "Converter PDF em Excel";

d) Abrir o arquivo de formato xlsx gerado pelo "Converter PDF em Excel"no Excel e exportar tabela por tabela em formato csv.

2.4 Etapas de limpeza dos dados

Devido a diferença entre os 3 datasets a serem analisados, o maior empenho do nosso trabalho foi nesta sessão.Foram desenvolvidos 3 métodos diferentes de limpeza(um para cada tabela) além de uma etapa final de limpeza, onde as tabelas passam todas a ter o nome dos campus de cada curso(não disponibilizado em algumas das tabelas), e por fim, em uma outra função, agregar tudo em uma tabela com os índices devidamente agrupados.

2.4.1 Preparação e Limpeza do primeiro dataset

O código da Figura 4 faz a limpeza do dataset que corresponde ao período do primeiro semestre de 2006 até o segundo semestre de 2007, ele faz a remoção do cabeçalho e do rodapé das tabelas, preenche a tabela campus com os resultados faltantes(o parâmetro method do método fillna quando vale "ffill"se comporta de forma a completar a linha com o primeiro valor valido imediatamente a cima), renomeia as colunas para facilitar o acesso e preenche as colunas faltantes, logo após isso ele faz a remoção da modalidade do curso do nome e cria uma tabela em separado para isso e por fim reindexa a tabela.

2.4.2 Preparação e Limpeza do segundo dataset.

O código da Figura 5 faz a limpeza do dataset que corresponde ao período do primeiro semestre de 2008 até o primeiro período de 2011, ele faz a remoção do cabeçalho e do rodapé das tabelas, quebra a coluna "Ocorrencia/Forma de evasão"em "tipo"e "quantidade"e em seguida faz o pivoteamento, ou seja, transforma a coluna tipo em varias colunas com os valores de dentro dela, atribuindo os valores nos devidos lugares e em seguida substituindo as células sem nenhum valor(NaN) por 0.

Figura 4: Código em Python para a limpeza do primeiro dataset.

preparação e limpeza do dataset 1

In [2]:

```
def repModalidade(data):
    data.reset_index(drop=True, inplace=True)
    if('modalidade' not in data.columns):
            data["modalidade"]='Não especificado'
    for i in range(len(data)):
        if(data.loc[i,('curso')].startswith('Licenciatura Em ')):
            data.loc[i,('modalidade')]='Licenciatura Plena'
            data.loc[i,('curso')]=data['curso'][i][16:]
        elif(data.loc[i,('curso')].endswith(' - Noturno')):
            data.loc[i,('curso')]=data['curso'][i][:-10]
        elif(data.loc[i,('curso')].endswith(' - Bacharelado')):
            data.loc[i,('modalidade')]='Bacharelado'
            data.loc[i,('curso')]=data['curso'][i][:-14]
        elif(data.loc[i,('curso')].endswith(' - Licenciatura Plena (Cod.1
            data.loc[i,('modalidade')]='Licenciatura Plena'
            data.loc[i,('curso')]=data['curso'][i][:-14]
        else:
            data.loc[i,('modalidade')]='Bacharelado'

def cleardbl(data):
    data.columns = data.loc[0]
    data.drop(data.index[:1], inplace=True)
    data.drop(data.index[-1], inplace=True)
    data["Campus"].fillna(method='ffill', inplace=True)
    data.rename({"NOME_CURSO": "curso", "Campus": "campus"}, axis=1, inpl
    data.curso = data.curso.str.title()
    data["turno"]='Não especificado'
    col = ["Abandono","Cancelamento","Classificado e Não Matriculado","De
    for i in col:
        if i not in data.columns:
            data[i] = np.nan
        data[i].fillna(0, inplace=True)

    data.dropna(axis=1, inplace=True)

    s=data['curso']
    s[s.str.startswith('Curso De ')]=s[s.str.startswith('Curso De ')].str
    data['curso']=s
    data['curso']=data['curso'].str.normalize('NFKD').str.encode('ascii',
    repModalidade(data)
    data.reset_index(drop=True, inplace=True)


#carregando as databases e limpando com os metodos definidos anteriorment
d061 = pd.read_csv("./data/evasao - 061.csv", encoding = "utf-8")
cleardbl(d061)
d061.drop(columns=["Nº de alunos"], inplace=True)

d062 = pd.read_csv("./data/evasao - 062.csv", encoding = "utf-8")
cleardbl(d062)
d062.drop(columns=["Nº de alunos"], inplace=True)

d071 = pd.read_csv("./data/evasao - 071.csv", encoding = "utf-8")
cleardbl(d071)
d071.drop(columns=["Nº alunos"], inplace=True)

d072 = pd.read_csv("./data/evasao - 072.csv", encoding = "utf-8")
cleardbl(d072)
d072.drop(columns=["Nº de alunos"], inplace=True)
```

Fonte: Próprios autores.

Figura 5: Código em Python para a limpeza do segundo dataset.

preparação e limpeza do dataset 2

In [3]:
```
def cleardb2(data, lastline):
    data.columns = data.loc[1]
    data.drop(data.index[:2], inplace=True)
    data.drop(data.index[lastline], inplace=True)
    data.drop(data.loc[data.NOME_CURSO=='Regime Especial De Graduacao'].i

    data["NOME_CURSO"].fillna(method='ffill', inplace=True)

    if ('Campus' in data.columns):
        data["Campus"].fillna(method='ffill', inplace=True)

    data.dropna(1, inplace=True)
    data.reset_index(drop=True, inplace=True)
    data = split(data)
    data = pivoting(data)
    data.curso = data.curso.str.title()
    s=data['curso']
    s[s.str.startswith('Curso De ')]=s[s.str.startswith('Curso De ')].str
    data['curso']=s
    data['curso']=data['curso'].str.normalize('NFKD').str.encode('ascii',
    repModalidade(data)
    return data

def split(data):
    ocorrencia = data[["Ocorrência/forma de evasão"]]
    ocorrencia = pd.DataFrame(ocorrencia["Ocorrência/forma de evasão"].st
    data = pd.concat([data, ocorrencia], axis=1, join_axes=[data.index])

    data.drop(columns=["Ocorrência/forma de evasão"], inplace=True)
    data.rename({"NOME_CURSO": "curso", 0:"tipo", 1:"quantidade"}, axis=1
    data.quantidade = data.quantidade.apply(pd.to_numeric)
    return data

def pivoting(data):
    data.dropna(inplace=True)
    data1 = data.pivot(index="curso", columns='tipo', values="quantidade"
    data1.fillna(0, inplace=True)
    data1.reset_index(inplace=True)
    data1.dropna(axis=0, inplace=True)
    return data1
```

In [4]:
```
#carregando as databases e limpando com os metodos definidos anteriorment
d081 = pd.read_csv("./data/evasao - 081.csv", encoding = "utf-8")
d081 = cleardb2(d081,-4)

d082 = pd.read_csv("./data/evasao - 082.csv", encoding = "utf-8")
d082 = cleardb2(d082, -5)

d091 = pd.read_csv("./data/evasao - 091.csv", encoding = "utf-8")
d091 = cleardb2(d091, -8)

d092 = pd.read_csv("./data/evasao - 092.csv", encoding = "utf-8")
d092 = cleardb2(d092, -6)

d101 = pd.read_csv("./data/evasao - 101.csv", encoding = "utf-8")
d101 = cleardb2(d101, -8)

d102 = pd.read_csv("./data/evasao - 102.csv", encoding = "utf-8")
d102 = cleardb2(d102,-7)

d111 = pd.read_csv("./data/evasao - 111.csv", encoding = "utf-8")
d111 = cleardb2(d111,-5)
```

Fonte: Próprios autores.

2.4.3 Preparação e Limpeza do terceiro dataset.

Devido ao formato da fonte do terceiro dataset, referente ao primeiro semestre de 2016 até o primeiro semestre de 2018, ser diferentes dos outros dois, foi necessário efetuar a limpeza e adaptação em 3 etapas, além de adotar algumas conveniências para que a adaptação seja feita corretamente.

As diferenças mais relevantes da fonte são:

1. Os arquivos CSV são disponibilizados mensalmente ao invés de semestralmente;
2. Os arquivos CSV não informam a quantidade de alunos que saíram da faculdade(independente do motivo), mas sim a quantidade de alunos matriculados em cada curso em cada mês;
3. Alguns cursos possuem nomes diferentes dependendo do mês e do ano.

O código da Figura 6 faz a leitura dos arquivos CSV e renomeia os cursos que apresentavam diferencia, adotando um nome especifico para cada um, assim eliminando, para todos os cursos que foram detectados com nome errôneo, o problema de número 3 citada anteriormente.

Na Figura 7 é feita a junção de cada dataset de acordo com o seu respectivo ano, colocando a referencia do mês/ano no formato "alunos matriculados em ano-mês", e logo após, os 3 datasets gerados são juntados em um único, para melhor manipulação na próxima etapa.

Porém, antes de iniciar a terceira etapa de limpeza do terceiro dataset, foram feitas algumas considerações para conseguir deixar o formato deste dataset o mais próximo dos outros dois:

- Será considero que o primeiro semestre do ano comece em Abril (pois é quando são contabilizados os alunos matriculados no mês de Fevereiro, Março e Abril) e termine em Julho (mês onde são contabilizados os alunos matriculados no final do período letivo), e que o segundo semestre comece em Agosto(mês em que as ferias terminam e ocorrem as transferências) e termine em Fevereiro (ultimo mês antes da contabilização das inscrições através do ENEM);
- O nível de precisão será menor do que das tabelas de antes de 2012, já que alunos podem ter abandonado os cursos ao mesmo tempo que novos entraram, assim mascarando a verdadeira quantidade de abandonos.
- O calculo para definir a quantidade de desistência de cada semestre será:

  $$\frac{\text{Quantidade de alunos no de inicio do semestre - Quantidade de alunos no de termino do semestre}}{}$$;

- Então, para o primeiro semestre sera:

  Quantidade de alunos matriculados em Abril - Quantidade de alunos matriculados em Julho;

- E para o segundo semestre será:

  Quantidade de alunos matriculados em Agosto - Quantidade de alunos matriculados em Fevereiro;

- Como os dados começam em Janeiro de 2016, os meses anteriores ao mês de Abril de 2016 serão removidos dos cálculos, ou seja, não será considerada a pequena parcela do segundo semestre de 2015;

- Como os dados da fonte vão apenas até o Abril de 2018, e não seria possível considerar a quantidade de desistências apenas com um mês, será considerado que o primeiro semestre de 2018 comece em Março e termine em Abril.

Então, no código presente na Figura 8, para cada semestre que desejamos analisar, é feita a diferença da quantidade de alunos matriculados entre o primeiro e o ultimo mês de cada curso, adicionando essa diferença a uma coluna do dataset, inicialmente chamada de "diff"mas renomeada para "abandono"e são removidas as colunas que representam os meses no dataset gerado. Então, todos os raros valores ausentes(NaN) e negativos são normalizados para 0, no intuito de evitar ao máximo o número de erros. E por ultimo, são adicionadas as colunas faltantes, é retiro a palava "Campus"do inicio dos nomes do campus e são retirados os acentos dos nomes dos cursos, para os dados ficarem iguais aos outros dois datasets.

Figura 6: Código em Python para a leitura dos CSV do terceiro dataset e renomeação dos cursos.

limpeza e preparação do dataset 3

In [5]:
```
a2016m01 = pd.read_csv("data/nAlunosPorMes/a2016m01.csv", encoding = "ISO-8859-1")
a2016m02 = pd.read_csv("data/nAlunosPorMes/a2016m02.csv", encoding = "ISO-8859-1")
a2016m03 = pd.read_csv("data/nAlunosPorMes/a2016m03.csv", encoding = "ISO-8859-1")
a2016m04 = pd.read_csv("data/nAlunosPorMes/a2016m04.csv", encoding = "ISO-8859-1")
a2016m05 = pd.read_csv("data/nAlunosPorMes/a2016m05.csv", encoding = "ISO-8859-1")
a2016m06 = pd.read_csv("data/nAlunosPorMes/a2016m06.csv", encoding = "ISO-8859-1")
a2016m07 = pd.read_csv("data/nAlunosPorMes/a2016m07.csv", encoding = "ISO-8859-1")
a2016m08 = pd.read_csv("data/nAlunosPorMes/a2016m08.csv", encoding = "ISO-8859-1")
a2016m10 = pd.read_csv("data/nAlunosPorMes/a2016m10.csv", encoding = "ISO-8859-1")
a2016m11 = pd.read_csv("data/nAlunosPorMes/a2016m11.csv", delimiter=';', encoding = "ISO-885
a2016m12 = pd.read_csv("data/nAlunosPorMes/a2016m12.csv", delimiter=';', encoding = "ISO-885

a2017m01 = pd.read_csv("data/nAlunosPorMes/a2017m01.csv", delimiter=';', encoding = "ISO-885
a2017m02 = pd.read_csv("data/nAlunosPorMes/a2017m02.csv", delimiter=';', encoding = "ISO-885
a2017m03 = pd.read_csv("data/nAlunosPorMes/a2017m03.csv", delimiter=';', encoding = "ISO-885
a2017m04 = pd.read_csv("data/nAlunosPorMes/a2017m04.csv", delimiter=',', encoding = "ISO-885
a2017m05 = pd.read_csv("data/nAlunosPorMes/a2017m05.csv", delimiter=',', encoding = "ISO-885
a2017m07 = pd.read_csv("data/nAlunosPorMes/a2017m07.csv", delimiter=',', encoding = "ISO-885
a2017m08 = pd.read_csv("data/nAlunosPorMes/a2017m08.csv", delimiter=',', encoding = "ISO-885
a2017m09 = pd.read_csv("data/nAlunosPorMes/a2017m09.csv", delimiter=',', encoding = "ISO-885
a2017m11 = pd.read_csv("data/nAlunosPorMes/a2017m11.csv", delimiter=',', encoding = "ISO-885
a2017m12 = pd.read_csv("data/nAlunosPorMes/a2017m12.csv", delimiter=',', encoding = "ISO-885

a2018m01 = pd.read_csv("data/nAlunosPorMes/a2018m01.csv", delimiter=',', encoding = "ISO-885
a2018m02 = pd.read_csv("data/nAlunosPorMes/a2018m02.csv", delimiter=',', encoding = "ISO-885
a2018m03 = pd.read_csv("data/nAlunosPorMes/a2018m03.csv", delimiter=',', encoding = "ISO-885
a2018m04 = pd.read_csv("data/nAlunosPorMes/a2018m04.csv", delimiter=',', encoding = "ISO-885

#colocando as referencias da lista para fazer uma primeira limpeza
datasets3=[a2016m01,a2016m02, a2016m03, a2016m04, a2016m05, a2016m06, a2016m07, a2016m08, a2
          a2017m01, a2017m02, a2017m03,a2017m04, a2017m05, a2017m07, a2017m08, a2017m09, a201
          a2018m01,a2018m02,a2018m03,a2018m04]

#constantes para jogar as tabelas em dataframes
a=['nome do curso','campus', 'nível', 'turno', 'modalidade',  'alunos matriculados']
b=['nome do curso', 'campus', 'nível', 'turno', 'modalidade']
```

In [6]:
```
find=['Letras -  Com Habilitação Em Lingua Portuguesa E Respectivas Literaturas',
      'Letras -  Com Habilitação Em Língua Portuguesa E Respectivas Literaturas',
     'Geografia Licenciatura', 'Licenciatura Em Educação Física', 'Licenciatura Em Pedagogia
     'Engenharia Em Agrimensura']
repl=['Letras Português']*2+['Licenciatura Em Geografia','Educação Física', 'Pedagogia', 'Eng

for i in range (len(datasets3)):
    datasets3[i]['nome do curso']=datasets3[i]['nome do curso'].str.title()
    datasets3[i]['nome do curso']=datasets3[i]['nome do curso'].str.strip()
    for j in range (len(find)):
        datasets3[i]['nome do curso']=datasets3[i]['nome do curso'].str.replace(find[j],repl

datasets3=None
```

Fonte: Próprios autores.

Figura 7: Código em Python para junção dos CSV lidos em datasets, separados por ano, e após, juntar todos os anos em um único dataset.

```
In [7]:
#Datasets 2016-01 a 2016-12
df1602=pd.merge(a2016m01[a],a2016m02[a], how='outer', on=b, suffixes=(' em 16-01',' em 16-02
df1603=pd.merge(df1602,a2016m03[a], how='outer', on=b)
df1604=pd.merge(df1603,a2016m04[a], how='outer', on=b, suffixes=(' em 16-03',' em 16-04'))
df1605=pd.merge(df1604,a2016m05[a], how='outer', on=b)
df1606=pd.merge(df1605,a2016m06[a], how='outer', on=b, suffixes=(' em 16-05',' em 16-06'))
df1607=pd.merge(df1606,a2016m07[a], how='outer', on=b)
df1608=pd.merge(df1607,a2016m08[a], how='outer', on=b, suffixes=(' em 16-07',' em 16-08'))
df1610=pd.merge(df1608,a2016m10[a], how='outer', on=b)
df1611=pd.merge(df1610,a2016m11[a], how='outer', on=b, suffixes=(' em 16-10',' em 16-11'))
df16final=pd.merge(df1611,a2016m11[a], how='outer', on=b)

df16final.drop(df16final.loc[df16final.nível!='Graduação'].index, inplace=True)
df16final.rename({'alunos matriculados': 'alunos matriculados em 16-12'}, axis=1, inplace=Tr
df16final.sort_values(by=['campus', 'nome do curso'], inplace=True, ascending=True)
df16final.reset_index(drop=True, inplace=True)

In [8]:
#Datasets 2017-01 a 2017-12
df1702=pd.merge(a2017m01[a],a2017m02[a], how='outer', on=b, suffixes=(' em 17-01',' em 17-02
df1703=pd.merge(df1702,a2017m03[a], how='outer', on=b)
df1704=pd.merge(df1703,a2017m04[a], how='outer', on=b, suffixes=(' em 17-03',' em 17-04'))
df1705=pd.merge(df1704,a2017m05[a], how='outer', on=b)
df1707=pd.merge(df1705,a2017m07[a], how='outer', on=b, suffixes=(' em 17-05',' em 17-07'))
df1708=pd.merge(df1707,a2017m08[a], how='outer', on=b)
df1709=pd.merge(df1708,a2017m09[a], how='outer', on=b, suffixes=(' em 17-08',' em 17-09'))
df1711=pd.merge(df1709,a2017m11[a], how='outer', on=b)
df17final=pd.merge(df1711,a2017m12[a], how='outer', on=b, suffixes=(' em 17-11',' em 17-12'))

df17final.drop(df17final.loc[df17final.nível!='Graduação'].index, inplace=True)
df17final.reset_index(drop=True, inplace=True)
df17final.sort_values(by=['campus', 'nome do curso'], inplace=True)

In [9]:
#Datasets 2018-01 a 2018-04
df1802=pd.merge(a2018m01[a],a2018m02[a], how='outer', on=b, suffixes=(' em 18-01',' em 18-02
df1803=pd.merge(df1802,a2018m03[a], how='outer', on=b)
df18final=pd.merge(df1803,a2018m04[a], how='outer', on=b, suffixes=(' em 18-03',' em 18-04'))

df18final.drop(df18final.loc[df18final.nível!='Graduação'].index, inplace=True)
df18final.reset_index(drop=True, inplace=True)

In [10]:
df1617=pd.merge(df16final,df17final, how='outer', on=b)
df1618=pd.merge(df1617,df18final, how='outer', on=b)
```

Fonte: Próprios autores.

Figura 8: Código em Python para a limpeza do terceiro dataset e separação do mesmo em semestres.

```
In [11]:
def dif(df1, mes1, mes2):
    retorno=df1.copy()
    pattern='alunos matriculados em '
    col1=pattern+mes1
    col2=pattern+mes2
    retorno['diff']=retorno[col1]-retorno[col2]
    return retorno[['nome do curso', 'campus','turno','modalidade', col1, col2, 'diff']]

def addcols(df):
    df['classificado e não matriculado']=0
    df['cancelamento']=0
    df['desligamento']=0
    df['falecimento']=0
    df['transferência']=0
    return df

def cleardb3(df, mes1, mes2):
    ab=dif(df, mes1, mes2)
    ab=ab[[c for c in ab.columns if 'alunos' not in c]]
    ab.rename({'diff': 'abandono'}, axis=1, inplace=True)
    ab.rename({'nome do curso': 'curso'}, axis=1, inplace=True)
    ab['abandono'].fillna(0, inplace=True)

    s=ab['abandono']
    s[s<0]=0
    ab['abandono']=s

    ab['campus'] = ab['campus'].str[7:]
    addcols(ab)
    ab.drop(ab.loc[ab.curso=='Aluno Em Regime Especial De Graduacao'].index, inplace=True)
    ab.reset_index(drop=True, inplace=True)
    ab['curso']=ab['curso'].str.normalize('NFKD').str.encode('ascii', errors='ignore').str.de
    return ab

df1=cleardb3(df1618,'16-04','16-07')
df2=cleardb3(df1618,'16-08','17-02')
df3=cleardb3(df1618,'17-04','17-07')
df4=cleardb3(df1618,'17-08','18-02')
df5=cleardb3(df1618,'18-03','18-04')
```

Fonte: Próprios autores.

2.4.4 Finalização da limpeza e Agregação dos datasets.

Antes de fazer a junção dos 3 tipos de datasets, todos datasets são adicionados em uma lista, para fazer o mapeamento dos campus e a renomeação das colunas. Ver Figura 9.

O mapeamento dos campus é feito utilizando um loop que percorre a lista com os datasets, e, para cada dataset que possui os campus, atualiza-se em um dicionário o nome do curso e o nome do campus. Caso este dicionario esteja vazio (quando o método é executado pela primeira vez), é feito um mapeamento utilizando como referencia os datasets do primeiro semestre de 2018 e segundo semestre de 2007, pois estes são os que mais possuem cursos(de acordo com a sua fonte).

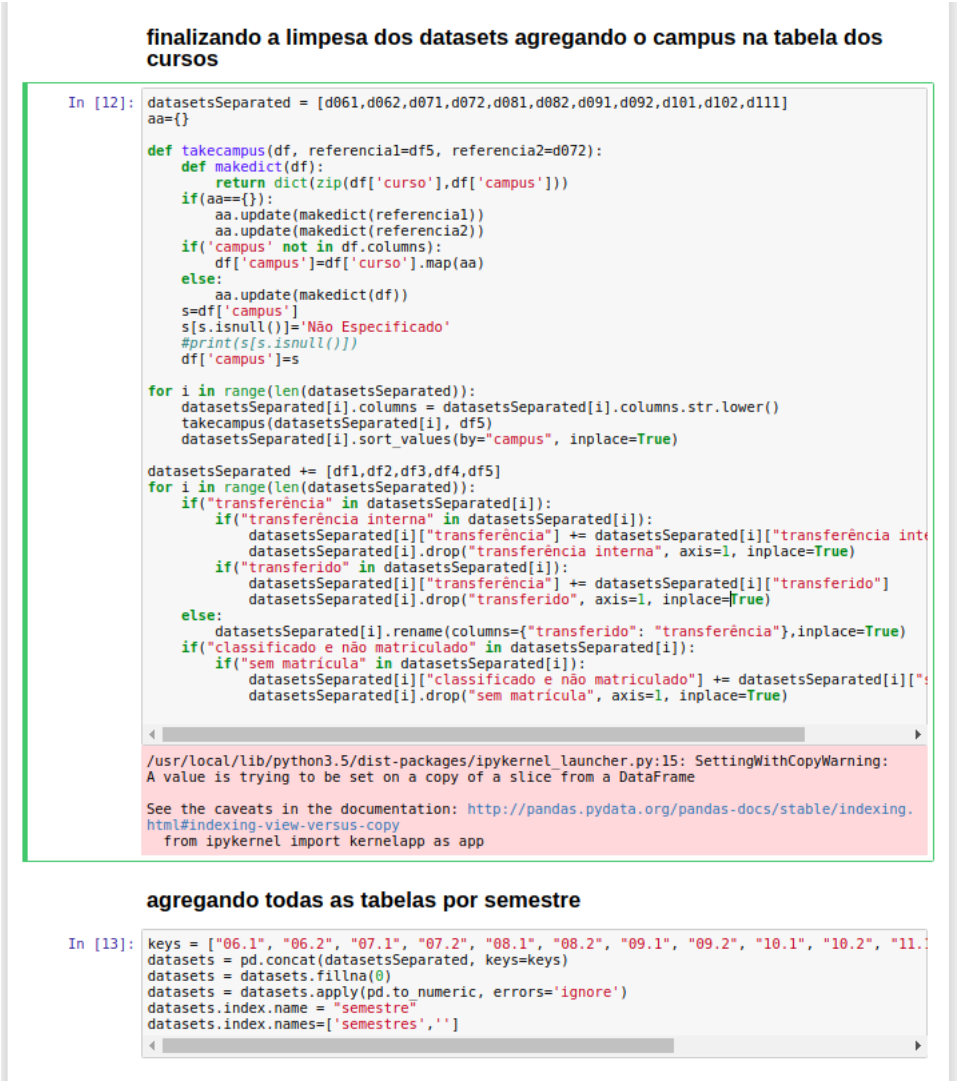
Caso o dataset não possua a coluna "campus", então esta é criada através do método map, passando como parâmetro o dicionario que contem os nomes dos campus de acordo com o curso. Caso um curso não seja encontrado dentro desse dicionario, o nome do campus será especificado como "Não especificado".

A renomeação das colunas é feita da seguinte maneira:

- Caso o dataset possua as colunas "transferência"e "transferência interna", os dados de "transferência interna"serão somados a "transferência"e "transferência interna"será apagada;
- Caso o dataset possua as colunas "transferência"e "transferido", os dados de "transferido"serão somados a "transferência"e "transferido"será apagada;
- Caso o dataset o dataset possua não possua a coluna "transferência", a coluna "transferido"será renomeada para "transferência";
- Caso o dataset possua a coluna "sem matrícula", os valores desta serão somados aos da coluna "classificado e não matriculado"e então será apagada.

Então todos os datasets serão juntados através do método concat da biblioteca pandas, passando como chaves para essa junção, os semestres (ex: 06.1, 17.2).

Figura 9: Código em Python para a finalização da limpeza e agregação dos datasets.



finalizando a limpesa dos datasets agregando o campus na tabela dos cursos

In [12]:

```
datasetsSeparated = [d061,d062,d071,d072,d081,d082,d091,d092,d101,d102,d111]
aa={}

def takecampus(df, referencial=df5, referencia2=d072):
    def makedict(df):
        return dict(zip(df['curso'],df['campus']))
    if(aa=={}):
        aa.update(makedict(referencial))
        aa.update(makedict(referencia2))
    if('campus' not in df.columns):
        df['campus']=df['curso'].map(aa)
    else:
        aa.update(makedict(df))
    s=df['campus']
    s[s.isnull()]='Não Especificado'
    #print(s[s.isnull()])
    df['campus']=s

for i in range(len(datasetsSeparated)):
    datasetsSeparated[i].columns = datasetsSeparated[i].columns.str.lower()
    takecampus(datasetsSeparated[i], df5)
    datasetsSeparated[i].sort_values(by="campus", inplace=True)

datasetsSeparated += [df1,df2,df3,df4,df5]
for i in range(len(datasetsSeparated)):
    if("transferência" in datasetsSeparated[i]):
        if("transferência interna" in datasetsSeparated[i]):
            datasetsSeparated[i]["transferência"] += datasetsSeparated[i]["transferência int
            datasetsSeparated[i].drop("transferência interna", axis=1, inplace=True)
        if("transferido" in datasetsSeparated[i]):
            datasetsSeparated[i]["transferência"] += datasetsSeparated[i]["transferido"]
            datasetsSeparated[i].drop("transferido", axis=1, inplace=True)
    else:
        datasetsSeparated[i].rename(columns={"transferido": "transferência"},inplace=True)
    if("classificado e não matriculado" in datasetsSeparated[i]):
        if("sem matrícula" in datasetsSeparated[i]):
            datasetsSeparated[i]["classificado e não matriculado"] += datasetsSeparated[i]["
            datasetsSeparated[i].drop("sem matrícula", axis=1, inplace=True)
```

```
/usr/local/lib/python3.5/dist-packages/ipykernel_launcher.py:15: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.
html#indexing-view-versus-copy
  from ipykernel import kernelapp as app
```

agregando todas as tabelas por semestre

In [13]:

```
keys = ["06.1", "06.2", "07.1", "07.2", "08.1", "08.2", "09.1", "09.2", "10.1", "10.2", "11.
datasets = pd.concat(datasetsSeparated, keys=keys)
datasets = datasets.fillna(0)
datasets = datasets.apply(pd.to_numeric, errors='ignore')
datasets.index.name = "semestre"
datasets.index.names=['semestres','']
```

Fonte: Próprios autores.

2.5 Visualização e Análise dos dados.

Nessa sessão serão respondidas as perguntas apresentadas na sessão 2.2 detalhadamente em forma de gráficos, listas e porcentagens, bem como algumas considerações sobre os datasets.

2.5.1 Considerações sobre os datasets.

Por conta da natureza diferente dos Datasets na consideração de algumas respostas alguns semestres foram desconsiderados, por exemplo, na comparação entre as categorias de evasão não foram considerados o terceiro dataset(2016 à 2018) pois nele não são discriminados essas categorias, ainda por conta de existir no terceiro dataset o turno dos cursos, diferentemente dos outros dois, apenas dele foram considerados os dados de evasão considerando os turnos.

2.5.2 Qual as cidades com o maior índice de evasões?.

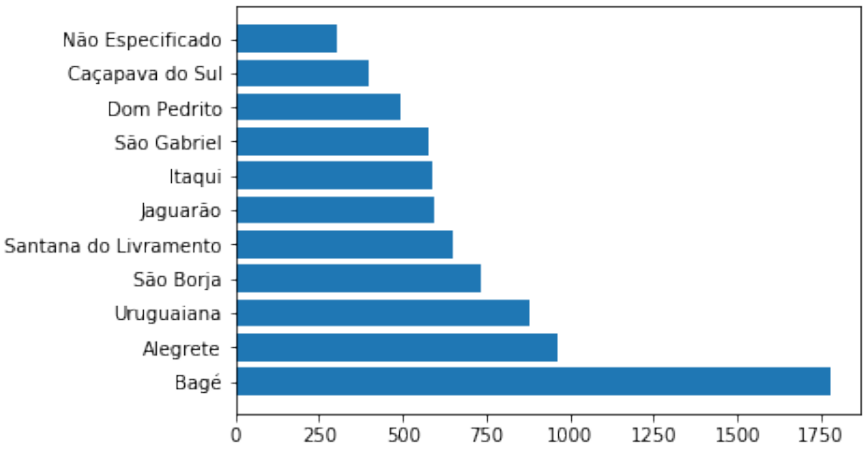
Para fazer as comparações entre as cidades foi preciso agrupar o dataset com base no campus e somar todas as ocorrências, não só isso, como também foi somado as colunas de categorias de evasão para fazer essa análise.

```
cidades = datasets.groupby("campus").sum().sum(axis=1).sort_values(ascending=False)
```

Com base nos dados do agrupamento podemos tiraras seguintes conclusões: O total de evasões no período análise se aproxima de 8 mil, sendo que 22% é de Bagé e 12% de Alegrete e ademais cidades no gráfico na figura 10.

```
cidades.sum()                          # sa da = 7956.0
cidades.Bage/cidades.sum()*100         # sa da = 22.37
cidades.Alegrete/cidades.sum()*100     # sa da = 12.06
```

Figura 10: Cidades com o maior índice de evasão.



Fonte: Próprios autores.

2.5.3 Qual a categoria de maior evasão?

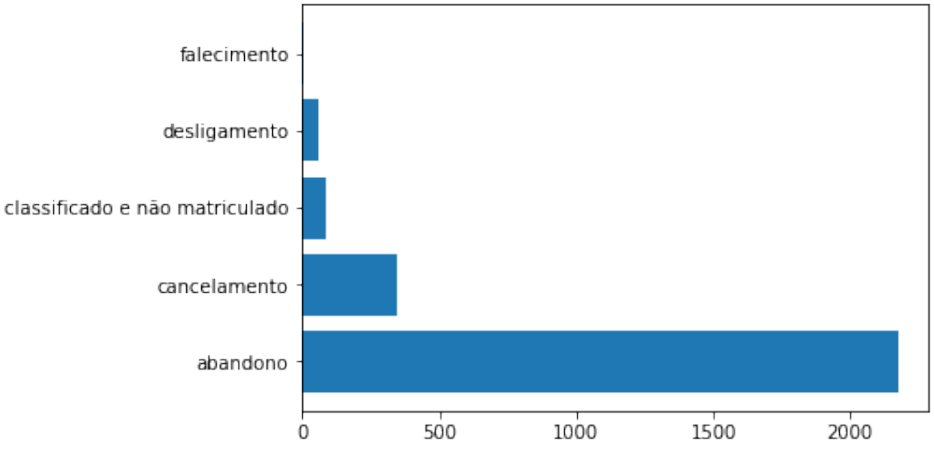
Para fazer as comparações entre as categorias foi preciso selecionar apenas as colunas de categorias e apenas os semestres que tem as categorias bem definidas e somalas.

```
categorias = ["abandono", "cancelamento", "classificado e n o matriculado",
              "desligamento","falecimento"]
cat = datasets[categorias]
cat = cat.loc[["06.1", "06.2", "07.1", "07.2", "08.1", "08.2", "09.1", "09.2", "10.1",
               "10.2", "11.1"]].sum()
```

A baixo podemos ver a lista de saída e o gráfico que ela gera, como podemos observar a maior causa de evasão da Unipampa dentro dos dados analisados se referem a abandono do curso e a menor delas é o falecimento.

```
cat
# saída:
# abandono                          2178.0
# cancelamento                       348.0
# classificado e n o matriculado      85.0
# desligamento                        59.0
# falecimento                          6.0
```

Figura 11: Categoria com o maior evasão.



Fonte: Próprios autores.

2.5.4 Qual o semestre de maior evasões?

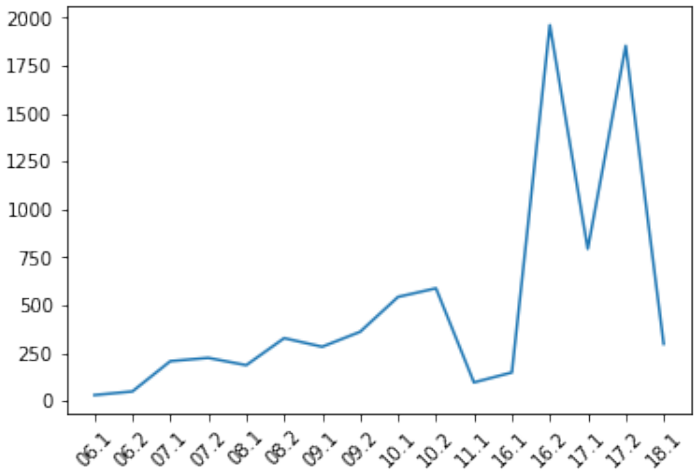
Para fazer as comparações entre as cidades foi preciso agrupar o dataset com base no semestre e somar todas as ocorrências, não só isso, como também foi necessário somar as colunas de categorias de evasão para fazer essa análise.

```
eps = datasets.groupby("semestres").sum().sum(axis=1)
```

Como pode ser visto na lista a baixo e na figura 12 a quantidade de abandonos tem aumentado drasticamente nos últimos anos, pode ser visto na figura 15 que apesar do número de cursos influenciar, não é o principal fator para este comportamento.

```
eps.sort_values(ascending=False).head()
# saida:
# 16.2    1961.0
# 17.2    1853.0
# 17.1     794.0
# 10.2     588.0
# 10.1     543.0
```

Figura 12: Evasão por semestre.



Fonte: Próprios autores.

2.5.5 Qual o curso de maior evasão?

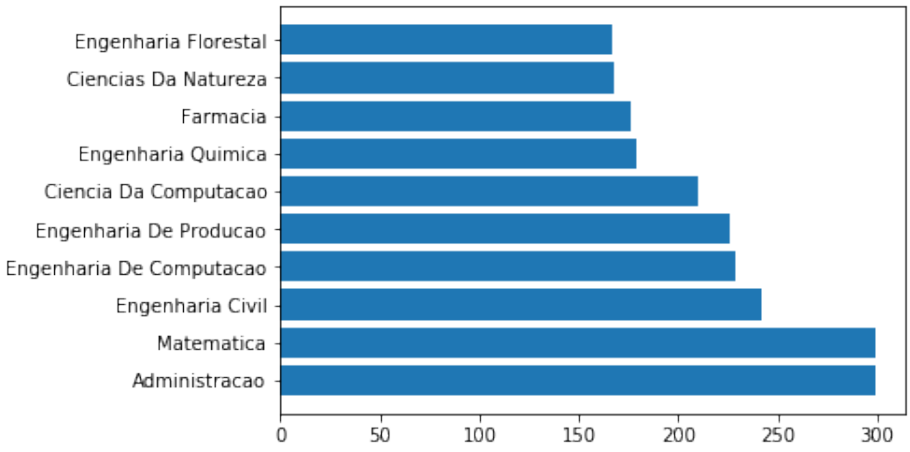
Da mesma forma com que foi feita a estatística das cidades só que agora agrupando pelos cursos e não pelos campus.

```
epc = datasets.groupby("curso").sum().sum(axis=1).sort_values(ascending=False)
```

Como pode ser visto na figura 13 e na lista a baixo os cursos com maior evasão, os 5 primeiros cursos sozinhos ocupam mais de 15% do total de evasão da universidade, considerando que no dataset existem um total de 95 curso, é um valor assustador.

```
epc.head()
# saida:
# Administracao                 299.0
# Matematica                    299.0
# Engenharia Civil              242.0
# Engenharia De Computacao      229.0
# Engenharia De Producao        226.0
(epc[0]+epc[1]+epc[2]+epc[3]+epc[4])/epc.sum()*100
# saida: 16.27%
epc.count()
# saida: 95
```

Figura 13: Cursos com o maior evasão.



Fonte: Próprios autores.

2.5.6 Qual o turno de curso com maior evasão?

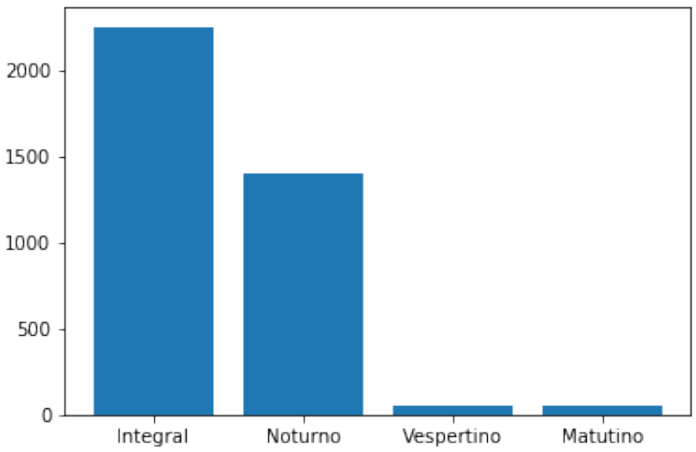
Para fazer a análise dos turnos, foi primeiro necessário agrupar os datasets pelo turno, e depois fazer a soma de todas as categorias de evasão.

```
turno = datasets.groupby("turno").count()
turno = turno.loc[["Integral", "Matutino", "Noturno", "Vespertino"]].sum(axis=1).
    sort_values(ascending=False)
```

Como pode ser visto, A esmagadora maioria das evasões da universidade se da nos cursos de turno integral e noturno, somando juntos 97% das evasões, como pode ser visto nos cálculos e na figura 14.

```
(turno["Integral"]+turno["Noturno"])/turno.sum()*100 # sa da = 97.33333333333334
turno["Integral"]/turno.sum()*100 # saida = 60.0
turno["Noturno"]/turno.sum()*100 # saida = 37.333
```

Figura 14: Turno com o maior evasão.



Fonte: Próprios autores.

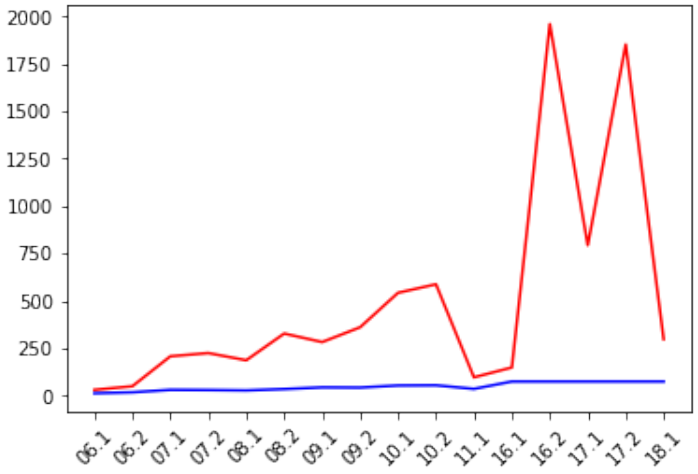
2.5.7 número de cursos em comparação ao número de evasões.

Para fazer essa comparação deve teve que ser feito um agrupamento pro semestre e uma contagem de quantos cursos e quantas evasões ocorreram por semestre.

```
nevasoes = datasets.groupby("semestres").sum().sum(axis=1)
ncursos = datasets.groupby("semestres")["curso"].count()
```

Nota-se por meio da figura 15 com o passar do tempo mais cursos são criados e maior o número de evasões, no entanto, o número de evasões cresce desproporcionalmente, aumentando quase que polinomialmente, conforme podemos notar tomando os segundos semestres dos anos.

Figura 15: número de cursos em comparação ao número de evasões.



Fonte: Próprios autores.

2.5.8 O maior número de evasões ocorre no primeiro ou no segundo semestre do ano?

Para fazer essa análise tivemos separar os dados em 2 categorias diferentes, primeiros e segundos semestres, para isso, desconsideramos os anos onde temos o primeiro e não temos o segundo semestre, para uma comparação mais justa.

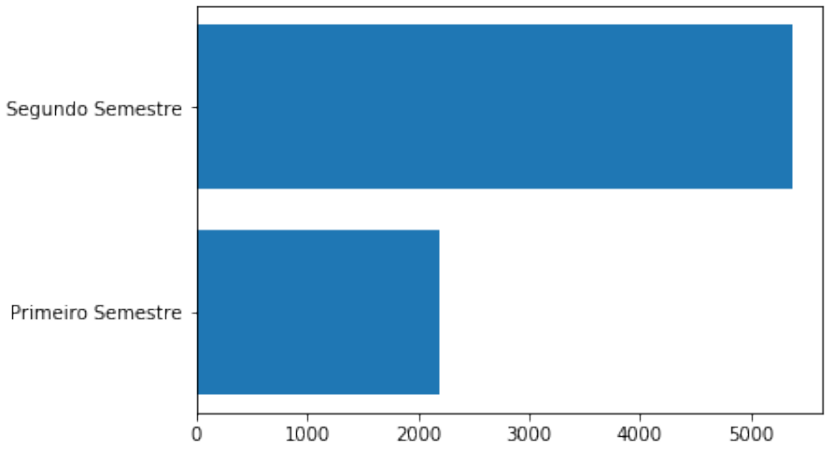
```
ss = ["06.2", "07.2", "08.2", "09.2", "10.2", "16.2", "17.2"]
ps = ["06.1", "07.1", "08.1", "09.1", "10.1", "16.1", "17.1"]
```

Após isso somamos todos os abandonos de cada uma das categorias e plotamos o gráfico da figura 16.

```
eps[ps].sum() # Primeiros Semestres - sa da = 2195.0
eps[ss].sum() # Segundos Semestres  - sa da = 5366.0
```

Com essa análise, podemos perceber que o número de evasões é muito maior no segundo do que no primeiro semestre de cada ano.

Figura 16: primeiro e segundo semestre do ano.



Fonte: Próprios autores.

3 Conclusão

Com base nas análises feitas a cima, pode-se concluir que o número de evasões é alarmante, o abandono tem sido a maior causa de evasão na Unipampa que cresse a cada ano. Grande Parte desses abandonos são advindos dos campus onde estão concentrados os cursos de engenharia, Bagé e Alegrete. Como pode-se ver na Figura 13, os cursos com maior evasão são os cursos que tem em seu currículo cadeiras de matemática avançada, como, por exemplo, Calculo I e II, Probabilidade e Estatística, Álgebra Linear, dentre outras. Nota-se também os cursos dos turno Integral e Noturno são os que possuem as maiores taxas de evasão.

Ainda à de se notar que os semestres de maior evasão nos cursos são os segundo semestres, como mostra a figura 16, oque pode indicar que os cursos tendem a ter suas cadeiras de maior dificuldade em semestres pares, que poderiam ser melhor distribuídos ao longo do curso.

Como Trabalhos futuros seria de grande interesse o aperfeiçoamento do código, a análise de datasets dos intervalos não analisados(segundo semestre de 2011 até o segundo semestre de 2015), se disponíveis e analisar mais profundamente os motivos do grande número de abandonos dos cursos da Unipampa.

Referências

JOSé, A. R.; ANDREOLI, G. S. 2011. Reitoria Unipampa, Divisao de apoio Pedagógico, Evasão. Disponível em: <http://porteiras.r.unipampa.edu.br/portais/cap/files/2010/07-/Relatório_final_evasão-na-UNIPAMPA_out20111.pdf>. Citado 3 vezes nas páginas 6, 7 e 8.

UNIPAMPA, D. de Tecnologia da Informação e C. 2018. Gestão unificada de recursos institucionais. Disponível em: <https://guri.unipampa.edu.br/rpt/relatorios-/dadosAbertos/>. Citado 3 vezes nas páginas 6, 7 e 8.

Apêndices

APÊNDICE A – Código Fonte Escrito para o Trabalho

Nesse apêndice encontra-se o código fonte escrito no ambiente Jupyter Notebook exportado para a linguagem Python puramente.

Arquivo: Trabalho.py

```
# coding: utf-8

## imports e declara  es

# In[1]:


#-*- coding: utf-8 -*-
import pandas as pd
import numpy as np
from matplotlib.pyplot import *


## Prepara  o, limpeza e agrega  o

## prepara  o e limpeza do dataset 1

# In[2]:


def repModalidade(data):
    data.reset_index(drop=True, inplace=True)
    if('modalidade' not in data.columns):
            data["modalidade"]='N o especificado'
    for i in range(len(data)):
        if(data.loc[i,('curso')].startswith('Licenciatura Em ')):
            data.loc[i,('modalidade')]='Licenciatura Plena'
            data.loc[i,('curso')]=data['curso'][i][16:]
        elif(data.loc[i,('curso')].endswith(' - Noturno')):
            data.loc[i,('curso')]=data['curso'][i][:-10]
        elif(data.loc[i,('curso')].endswith(' - Bacharelado')):
            data.loc[i,('modalidade')]='Bacharelado'
            data.loc[i,('curso')]=data['curso'][i][:-14]
        elif(data.loc[i,('curso')].endswith(' - Licenciatura Plena (Cod.111.Uni)')):
            data.loc[i,('modalidade')]='Licenciatura Plena'
            data.loc[i,('curso')]=data['curso'][i][:-14]
        else:
            data.loc[i,('modalidade')]='Bacharelado'

def cleardb1(data):
    data.columns = data.loc[0]
    data.drop(data.index[:1], inplace=True)
    data.drop(data.index[-1], inplace=True)
    data["Campus"].fillna(method='ffill', inplace=True)
```

```
    data.rename({"NOME_CURSO": "curso", "Campus": "campus"}, axis=1, inplace=True)
    data.curso = data.curso.str.title()
    data["turno"]='N o especificado'
    col = ["Abandono","Cancelamento","Classificado e N o Matriculado","Desligamento","
        Transferido"]
    for i in col:
        if i not in data.columns:
            data[i] = np.nan
        data[i].fillna(0, inplace=True)

    data.dropna(axis=1, inplace=True)

    s=data['curso']
    s[s.str.startswith('Curso De ')]=s[s.str.startswith('Curso De ')].str[9:]
    data['curso']=s
    data['curso']=data['curso'].str.normalize('NFKD').str.encode('ascii', errors='ignore'
        ).str.decode('utf-8')
    repModalidade(data)
    data.reset_index(drop=True, inplace=True)


#carregando as databases e limpando com os metodos definidos anteriormente
d061 = pd.read_csv("./data/evasao - 061.csv", encoding = "utf-8")
cleardb1(d061)
d061.drop(columns=["N de alunos"], inplace=True)

d062 = pd.read_csv("./data/evasao - 062.csv", encoding = "utf-8")
cleardb1(d062)
d062.drop(columns=["N de alunos"], inplace=True)

d071 = pd.read_csv("./data/evasao - 071.csv", encoding = "utf-8")
cleardb1(d071)
d071.drop(columns=["N alunos"], inplace=True)

d072 = pd.read_csv("./data/evasao - 072.csv", encoding = "utf-8")
cleardb1(d072)
d072.drop(columns=["N de alunos"], inplace=True)


### prepara  o e limpeza do dataset 2

# In[3]:


def cleardb2(data, lastline):
    data.columns = data.loc[1]
    data.drop(data.index[:2], inplace=True)
    data.drop(data.index[lastline], inplace=True)
    data.drop(data.loc[data.NOME_CURSO=='Regime Especial De Graduacao'].index, inplace=
        True)

    data["NOME_CURSO"].fillna(method='ffill', inplace=True)

    if ('Campus' in data.columns):
        data["Campus"].fillna(method='ffill', inplace=True)

    data.dropna(1, inplace=True)
    data.reset_index(drop=True, inplace=True)
    data = split(data)
    data = pivoting(data)
```

```
    data.curso = data.curso.str.title()
    s=data['curso']
    s[s.str.startswith('Curso De ')]=s[s.str.startswith('Curso De ')].str[9:]
    data['curso']=s
    data['curso']=data['curso'].str.normalize('NFKD').str.encode('ascii', errors='ignore'
        ).str.decode('utf-8')
    repModalidade(data)
    return data

def split(data):
    ocorrencia = data[["Ocorr ncia/forma de evas o"]]
    ocorrencia = pd.DataFrame(ocorrencia["Ocorr ncia/forma de evas o"].str.split("=").
        tolist())
    data = pd.concat([data, ocorrencia], axis=1, join_axes=[data.index])

    data.drop(columns=["Ocorr ncia/forma de evas o"], inplace=True)
    data.rename({"NOME_CURSO": "curso", 0:"tipo", 1:"quantidade"}, axis=1, inplace=True)
    data.quantidade = data.quantidade.apply(pd.to_numeric)
    return data

def pivoting(data):
    data.dropna(inplace=True)
    data1 = data.pivot(index="curso", columns='tipo', values="quantidade")
    data1.fillna(0, inplace=True)
    data1.reset_index(inplace=True)
    data1.dropna(axis=0, inplace=True)
    return data1


# In[4]:


#carregando as databases e limpando com os metodos definidos anteriormente
d081 = pd.read_csv("./data/evasao - 081.csv", encoding = "utf-8")
d081 = cleardb2(d081,-4)

d082 = pd.read_csv("./data/evasao - 082.csv", encoding = "utf-8")
d082 = cleardb2(d082, -5)

d091 = pd.read_csv("./data/evasao - 091.csv", encoding = "utf-8")
d091 = cleardb2(d091, -8)

d092 = pd.read_csv("./data/evasao - 092.csv", encoding = "utf-8")
d092 = cleardb2(d092, -6)

d101 = pd.read_csv("./data/evasao - 101.csv", encoding = "utf-8")
d101 = cleardb2(d101, -8)

d102 = pd.read_csv("./data/evasao - 102.csv", encoding = "utf-8")
d102 = cleardb2(d102,-7)

d111 = pd.read_csv("./data/evasao - 111.csv", encoding = "utf-8")
d111 = cleardb2(d111,-5)




## limpeza e prepara  o do dataset 3

# In[5]:
```

```
a2016m01 = pd.read_csv("data/nAlunosPorMes/a2016m01.csv", encoding = "ISO-8859-1")
a2016m02 = pd.read_csv("data/nAlunosPorMes/a2016m02.csv", encoding = "ISO-8859-1")
a2016m03 = pd.read_csv("data/nAlunosPorMes/a2016m03.csv", encoding = "ISO-8859-1")
a2016m04 = pd.read_csv("data/nAlunosPorMes/a2016m04.csv", encoding = "ISO-8859-1")
a2016m05 = pd.read_csv("data/nAlunosPorMes/a2016m05.csv", encoding = "ISO-8859-1")
a2016m06 = pd.read_csv("data/nAlunosPorMes/a2016m06.csv", encoding = "ISO-8859-1")
a2016m07 = pd.read_csv("data/nAlunosPorMes/a2016m07.csv", encoding = "ISO-8859-1")
a2016m08 = pd.read_csv("data/nAlunosPorMes/a2016m08.csv", encoding = "ISO-8859-1")
a2016m10 = pd.read_csv("data/nAlunosPorMes/a2016m10.csv", encoding = "ISO-8859-1")
a2016m11 = pd.read_csv("data/nAlunosPorMes/a2016m11.csv", delimiter=';', encoding = "ISO
    -8859-1")
a2016m12 = pd.read_csv("data/nAlunosPorMes/a2016m12.csv", delimiter=';', encoding = "ISO
    -8859-1")

a2017m01 = pd.read_csv("data/nAlunosPorMes/a2017m01.csv", delimiter=';', encoding = "ISO
    -8859-1")
a2017m02 = pd.read_csv("data/nAlunosPorMes/a2017m02.csv", delimiter=';', encoding = "ISO
    -8859-1")
a2017m03 = pd.read_csv("data/nAlunosPorMes/a2017m03.csv", delimiter=';', encoding = "ISO
    -8859-1")
a2017m04 = pd.read_csv("data/nAlunosPorMes/a2017m04.csv", delimiter=',', encoding = "ISO
    -8859-1")
a2017m05 = pd.read_csv("data/nAlunosPorMes/a2017m05.csv", delimiter=',', encoding = "ISO
    -8859-1")
a2017m07 = pd.read_csv("data/nAlunosPorMes/a2017m07.csv", delimiter=',', encoding = "ISO
    -8859-1")
a2017m08 = pd.read_csv("data/nAlunosPorMes/a2017m08.csv", delimiter=',', encoding = "ISO
    -8859-1")
a2017m09 = pd.read_csv("data/nAlunosPorMes/a2017m09.csv", delimiter=',', encoding = "ISO
    -8859-1")
a2017m11 = pd.read_csv("data/nAlunosPorMes/a2017m11.csv", delimiter=',', encoding = "ISO
    -8859-1")
a2017m12 = pd.read_csv("data/nAlunosPorMes/a2017m12.csv", delimiter=',', encoding = "ISO
    -8859-1")

a2018m01 = pd.read_csv("data/nAlunosPorMes/a2018m01.csv", delimiter=',', encoding = "ISO
    -8859-1")
a2018m02 = pd.read_csv("data/nAlunosPorMes/a2018m02.csv", delimiter=',', encoding = "ISO
    -8859-1")
a2018m03 = pd.read_csv("data/nAlunosPorMes/a2018m03.csv", delimiter=',', encoding = "ISO
    -8859-1")
a2018m04 = pd.read_csv("data/nAlunosPorMes/a2018m04.csv", delimiter=',', encoding = "ISO
    -8859-1")

#colocando as referencias da lista para fazer uma primeira limpeza
datasets3=[a2016m01,a2016m02, a2016m03, a2016m04, a2016m05, a2016m06, a2016m07, a2016m08,
     a2016m10, a2016m11, a2016m12,
          a2017m01, a2017m02, a2017m03,a2017m04, a2017m05, a2017m07, a2017m08, a2017m09,
              a2017m11, a2017m12,
          a2018m01,a2018m02,a2018m03,a2018m04]

#constantes para jogar as tabelas em dataframes
a=['nome do curso','campus', 'n vel', 'turno', 'modalidade', 'alunos matriculados']
b=['nome do curso', 'campus', 'n vel', 'turno', 'modalidade']


# In[6]:
```

```
find=['Letras␣-␣␣Com␣Habilita  o␣Em␣Lingua␣Portuguesa␣E␣Respectivas␣Literaturas',
      'Letras␣-␣␣Com␣Habilita  o␣Em␣L ngua␣Portuguesa␣E␣Respectivas␣Literaturas',
     'Geografia␣Licenciatura', 'Licenciatura␣Em␣Educa  o␣F sica', 'Licenciatura␣Em␣
         Pedagogia',
     'Engenharia␣Em␣Agrimensura']
repl=['Letras␣Portugu s']*2+['Licenciatura␣Em␣Geografia','Educa  o␣F sica', '
    Pedagogia', 'Engenharia␣De␣Agrimensura']

for i in range (len(datasets3)):
    datasets3[i]['nome␣do␣curso']=datasets3[i]['nome␣do␣curso'].str.title()
    datasets3[i]['nome␣do␣curso']=datasets3[i]['nome␣do␣curso'].str.strip()
    for j in range (len(find)):
        datasets3[i]['nome␣do␣curso']=datasets3[i]['nome␣do␣curso'].str.replace(find[j],
            repl[j])

datasets3=None


# In[7]:


#Datasets 2016-01 a 2016-12
df1602=pd.merge(a2016m01[a],a2016m02[a], how='outer', on=b, suffixes=('␣em␣16-01','␣em␣
    16-02'))
df1603=pd.merge(df1602,a2016m03[a], how='outer', on=b)
df1604=pd.merge(df1603,a2016m04[a], how='outer', on=b, suffixes=('␣em␣16-03','␣em␣16-04')
    )
df1605=pd.merge(df1604,a2016m05[a], how='outer', on=b)
df1606=pd.merge(df1605,a2016m06[a], how='outer', on=b, suffixes=('␣em␣16-05','␣em␣16-06')
    )
df1607=pd.merge(df1606,a2016m07[a], how='outer', on=b)
df1608=pd.merge(df1607,a2016m08[a], how='outer', on=b, suffixes=('␣em␣16-07','␣em␣16-08')
    )
df1610=pd.merge(df1608,a2016m10[a], how='outer', on=b)
df1611=pd.merge(df1610,a2016m11[a], how='outer', on=b, suffixes=('␣em␣16-10','␣em␣16-11')
    )
df16final=pd.merge(df1611,a2016m11[a], how='outer', on=b)

df16final.drop(df16final.loc[df16final.n vel!='Gradua  o'].index, inplace=True)
df16final.rename({'alunos␣matriculados': 'alunos␣matriculados␣em␣16-12'}, axis=1, inplace
    =True)
df16final.sort_values(by=['campus', 'nome␣do␣curso'], inplace=True, ascending=True)
df16final.reset_index(drop=True, inplace=True)


# In[8]:


#Datasets 2017-01 a 2017-12
df1702=pd.merge(a2017m01[a],a2017m02[a], how='outer', on=b, suffixes=('␣em␣17-01','␣em␣
    17-02'))
df1703=pd.merge(df1702,a2017m03[a], how='outer', on=b)
df1704=pd.merge(df1703,a2017m04[a], how='outer', on=b, suffixes=('␣em␣17-03','␣em␣17-04')
    )
df1705=pd.merge(df1704,a2017m05[a], how='outer', on=b)
df1707=pd.merge(df1705,a2017m07[a], how='outer', on=b, suffixes=('␣em␣17-05','␣em␣17-07')
    )
df1708=pd.merge(df1707,a2017m08[a], how='outer', on=b)
df1709=pd.merge(df1708,a2017m09[a], how='outer', on=b, suffixes=('␣em␣17-08','␣em␣17-09')
```

```
    )
df1711=pd.merge(df1709,a2017m11[a], how='outer', on=b)
df17final=pd.merge(df1711,a2017m12[a], how='outer', on=b, suffixes=('␣em␣17-11','␣em␣
    17-12'))

df17final.drop(df17final.loc[df17final.n vel!='Gradua  o'].index, inplace=True)
df17final.reset_index(drop=True, inplace=True)
df17final.sort_values(by=['campus', 'nome␣do␣curso'], inplace=True)


# In[9]:


#Datasets 2018-01 a 2018-04
df1802=pd.merge(a2018m01[a],a2018m02[a], how='outer', on=b, suffixes=('␣em␣18-01','␣em␣
    18-02'))
df1803=pd.merge(df1802,a2018m03[a], how='outer', on=b)
df18final=pd.merge(df1803,a2018m04[a], how='outer', on=b, suffixes=('␣em␣18-03','␣em␣
    18-04'))

df18final.drop(df18final.loc[df18final.n vel!='Gradua  o'].index, inplace=True)
df18final.reset_index(drop=True, inplace=True)


# In[10]:


df1617=pd.merge(df16final,df17final, how='outer', on=b)
df1618=pd.merge(df1617,df18final, how='outer', on=b)


# In[11]:


def dif(df1, mes1, mes2):
    retorno=df1.copy()
    pattern='alunos␣matriculados␣em␣'
    col1=pattern+mes1
    col2=pattern+mes2
    retorno['diff']=retorno[col1]-retorno[col2]
    return retorno[['nome␣do␣curso', 'campus','turno','modalidade', col1, col2, 'diff']]

def addcols(df):
    df['classificado␣e␣n o␣matriculado']=0
    df['cancelamento']=0
    df['desligamento']=0
    df['falecimento']=0
    df['transfer ncia']=0
    return df

def cleardb3(df, mes1, mes2):
    ab=dif(df, mes1, mes2)
    ab=ab[[c for c in ab.columns if 'alunos' not in c]]
    ab.rename({'diff': 'abandono'}, axis=1, inplace=True)
    ab.rename({'nome␣do␣curso': 'curso'}, axis=1, inplace=True)
    ab['abandono'].fillna(0, inplace=True)

    s=ab['abandono']
    s[s<0]=0
    ab['abandono']=s
```

```
    ab['campus'] = ab['campus'].str[7:]
    addcols(ab)
    ab.drop(ab.loc[ab.curso=='Aluno Em Regime Especial De Graduacao'].index, inplace=True
        )
    ab.reset_index(drop=True, inplace=True)
    ab['curso']=ab['curso'].str.normalize('NFKD').str.encode('ascii', errors='ignore').
        str.decode('utf-8')
    return ab

df1=cleardb3(df1618,'16-04','16-07')
df2=cleardb3(df1618,'16-08','17-02')
df3=cleardb3(df1618,'17-04','17-07')
df4=cleardb3(df1618,'17-08','18-02')
df5=cleardb3(df1618,'18-03','18-04')


### finalizando a limpesa dos datasets agregando o campus na tabela dos cursos

# In[12]:


datasetsSeparated = [d061,d062,d071,d072,d081,d082,d091,d092,d101,d102,d111]
aa={}

def takecampus(df, referencia1=df5, referencia2=d072):
    def makedict(df):
        return dict(zip(df['curso'],df['campus']))
    if(aa=={}):
        aa.update(makedict(referencia1))
        aa.update(makedict(referencia2))
    if('campus' not in df.columns):
        df['campus']=df['curso'].map(aa)
    else:
        aa.update(makedict(df))
    s=df['campus']
    s[s.isnull()]='N o Especificado'
    #print(s[s.isnull()])
    df['campus']=s

for i in range(len(datasetsSeparated)):
    datasetsSeparated[i].columns = datasetsSeparated[i].columns.str.lower()
    takecampus(datasetsSeparated[i], df5)
    datasetsSeparated[i].sort_values(by="campus", inplace=True)

datasetsSeparated += [df1,df2,df3,df4,df5]
for i in range(len(datasetsSeparated)):
    if("transfer ncia" in datasetsSeparated[i]):
        if("transfer ncia interna" in datasetsSeparated[i]):
            datasetsSeparated[i]["transfer ncia"] += datasetsSeparated[i]["
                transfer ncia interna"]
            datasetsSeparated[i].drop("transfer ncia interna", axis=1, inplace=True)
        if("transferido" in datasetsSeparated[i]):
            datasetsSeparated[i]["transfer ncia"] += datasetsSeparated[i]["transferido"]
            datasetsSeparated[i].drop("transferido", axis=1, inplace=True)
    else:
        datasetsSeparated[i].rename(columns={"transferido": "transfer ncia"},inplace=
            True)
    if("classificado e n o matriculado" in datasetsSeparated[i]):
        if("sem matr cula" in datasetsSeparated[i]):
```

```
            datasetsSeparated[i]["classificado␣e␣n o␣matriculado"] += datasetsSeparated[
                i]["sem␣matr cula"]
            datasetsSeparated[i].drop("sem␣matr cula", axis=1, inplace=True)


### agregando todas as tabelas por semestre

# In[13]:


keys = ["06.1", "06.2", "07.1", "07.2", "08.1", "08.2", "09.1", "09.2", "10.1", "10.2", "
    11.1", "16.1", "16.2", "17.1", "17.2", "18.1"]
datasets = pd.concat(datasetsSeparated, keys=keys)
datasets = datasets.fillna(0)
datasets = datasets.apply(pd.to_numeric, errors='ignore')
datasets.index.name = "semestre"
datasets.index.names=['semestres','']


## Respostas para as perguntas:

### Qual as cidades com o maior indice de evas es?

# In[14]:


cidades = datasets.groupby("campus").sum().sum(axis=1).sort_values(ascending=False)
cidades.head()


# In[15]:


cidades.sum()


# In[16]:


cidades.Bag /cidades.sum()*100


# In[17]:


cidades.Alegrete/cidades.sum()*100


# In[18]:


barh(cidades.index, cidades.values)


### Qual a categoria de maior evas o?

# In[19]:


categorias = ["abandono", "cancelamento", "classificado␣e␣n o␣matriculado", "
```

```
    desligamento","falecimento"]
cat = datasets[categorias]
cat = cat.loc[["06.1", "06.2", "07.1", "07.2", "08.1", "08.2", "09.1", "09.2", "10.1", "
    10.2", "11.1"]].sum()


# In[20]:


cat


# In[21]:


barh(cat.index, cat.values)


# In[22]:


cat.loc["abandono"]/cat.sum()*100


### Qual o semestre de maior evas es?

# In[23]:


eps = datasets.groupby("semestres").sum().sum(axis=1)


# In[24]:


eps.sort_values(ascending=False).head()


# In[25]:


xticks(rotation=45)
plot(eps.index, eps.values)


### Qual o curso de maior evas o?

# In[26]:


epc = datasets.groupby("curso").sum().sum(axis=1).sort_values(ascending=False)


# In[27]:


epc.head(10)


# In[28]:
```

```
barh(epc.head(10).index, epc.head(10).values)

# In[29]:

(epc[0]+epc[1]+epc[2]+epc[3]+epc[4])/epc.sum()*100

# In[30]:

epc.count()

## Qual o turno de curso com maior evas o?

# In[31]:

turno = datasets.groupby("turno").count()
turno = turno.loc[["Integral", "Matutino", "Noturno", "Vespertino"]].sum(axis=1).
    sort_values(ascending=False)

# In[32]:

bar(turno.index, turno.values)

# In[33]:

(turno["Integral"]+turno["Noturno"])/turno.sum()*100

# In[34]:

turno["Integral"]/turno.sum()*100

# In[35]:

turno["Noturno"]/turno.sum()*100

## Numero de cursos em compara  o ao numero de evas es

# In[36]:

nevasoes = datasets.groupby("semestres").sum().sum(axis=1)
ncursos = datasets.groupby("semestres")["curso"].count()
```

```
# In[37]:


xticks(rotation=45)
plot(ncursos.index, ncursos.values, color="blue")#azul=cursos
plot(ncursos.index, nevasoes.values, color="red")#vermelho=evasoes


### O maior numero de evasoes ocorre no primeiro ou mo segundo semestredo ano?

# In[38]:


ss = ["06.2", "07.2", "08.2", "09.2", "10.2", "16.2", "17.2"]
ps = ["06.1", "07.1", "08.1", "09.1", "10.1", "16.1", "17.1"]


# In[39]:


eps[ss].sum()


# In[40]:


eps[ps].sum()


# In[41]:


barh(["Primeiro Semestre", "Segundo Semestre"], [eps[ps].sum(), eps[ss].sum()])


# In[42]:


eps[ss].sum()/eps[ps+ss].sum()*100


# In[43]:


eps[ps].sum()/eps[ps+ss].sum()*100
```

Anexos

ANEXO A – Dataset número 1

Nesse Anexo estão linkadas todas as tabelas do dataset 1, ja devidamente extraidas do pdf.

A.1 Primeiro semestre de 2006

https://drive.google.com/file/d/1qfwDgLi3rLTpe9Y8j44V45f_q8vW5a3C/view?usp=sharing

A.2 Segundo semestre de 2006

https://drive.google.com/file/d/1lNMDJl69GOht0mAnLISYCP1xiwj8lc1e/view?usp=sharing

A.3 Primeiro semestre de 2007

https://drive.google.com/file/d/1cdWB3T8uhf2Jk_6U186e7GilhytxPkJ1/view?usp=sharing

A.4 Segundo semestre de 2007

https://drive.google.com/file/d/1NLSc1MOg1MLtpQ36CgVjqlzw5X5E-eYk/view?usp=sharing

ANEXO B – Dataset número 2

Nesse Anexo estão linkadas todas as tabelas do dataset 2, ja devidamente extraidas do pdf.

B.1 Primeiro semestre de 2008

https://drive.google.com/file/d/1eu4dW_i5iiGV_bHmOLYbjvKWztTYwW3I/view?usp=sharing

B.2 Segundo semestre de 2008

https://drive.google.com/file/d/1cVe2MlDmjC9JO7pswSYnuvfPaIUBYg94/view?usp=sharing

B.3 Primeiro semestre de 2009

https://drive.google.com/file/d/1uD3ivHV2IO5WIdkSAdrd5TbHSBb9yDOO/view?usp=sharing

B.4 Segundo semestre de 2009

https://drive.google.com/file/d/1_A7PoW7Oj21s9s3Q9sXa3O5T3eHlBlJ-/view?usp=sharing

B.5 Primeiro semestre de 2010

https://drive.google.com/file/d/16M1G5KPi5Ov8DEPh2rDc4kPfeSCgjHjK/view?usp=sharing

B.6 Segundo semestre de 2010

https://drive.google.com/file/d/144X54E5wvWjU1M1LnOfOKoYRsDTrx9mY/view?usp=sharing

B.7 Primeiro semestre de 2011

https://drive.google.com/file/d/1RBxTd5Bf3CSsSYxzXKRrzZGLAAI5o7wV/view?usp=sharing

ANEXO C – Dataset número 3

Nesse Anexo estão linkadas todas as tabelas do dataset 1

C.1 01.2016

https://drive.google.com/file/d/1H3ZVftpB5kUz5moEi3mZEPgk6C8kZgjs/view?usp=sharing

C.2 02.2016

https://drive.google.com/file/d/1TTifSZeFPm9UeZ347ksPHp36wtZeyIiX/view?usp=sharing

C.3 03.2016

https://drive.google.com/file/d/1qloX9AMtEvY-q5UqbAijmyxDIs7fAwtO/view?usp=sharing

C.4 04.2016

https://drive.google.com/file/d/1O2vqSIj3oVz07WqLGn9tdL634ICCNf0n/view?usp=sharing

C.5 05.2016

https://drive.google.com/file/d/1VXBJshXBpI4A4s1TRW7RapusPbiQ4c_6/view?usp=sharing

C.6 06.2016

https://drive.google.com/file/d/1JQslstn0tOqzAS4eLyBQ9VTEZKeEi6UT/view?usp=sharing

C.7 07.2016

https://drive.google.com/file/d/1H7raoDEl-dobZbijMgJz-QvubyfVlvf4/view?usp=sharing

C.8 08.2016

https://drive.google.com/file/d/1CQJfo85w1CtqaCBPljQmYcp6K00PHtVe/view?usp=sharing

C.9 10.2016

https://drive.google.com/file/d/1aj96Dg7syRSPnw8Xd3pSpB08M67RJBge/view?usp=sharing

C.10 11.2016

https://drive.google.com/file/d/1pE3zdpg5sn-w3EPIoJfGC6SvXo5sgI94/view?usp=sharing

C.11 12.2016

https://drive.google.com/file/d/1I79udVmCu0yrweunaCSmGZDD2wk2yVt9/view?usp=sharing

C.12 01.2017

https://drive.google.com/file/d/1UbHUTfJl_v1BPDS7TiI-LPB2zM3SdwS3/view?usp=sharing

C.13 02.2017

https://drive.google.com/file/d/1pNP1TIRJsjl_4B-UIKmPnkDdTUJEVrgo/view?usp=sharing

C.14 03.2017

https://drive.google.com/file/d/1kCWTScZxNMuJlMlTdW_XbEvOck_NafqK/view?usp=sharing

C.15 04.2017

https://drive.google.com/file/d/10YpbiroT4Hyr7wm0ODT1G1F1DWq5mrG3/view?usp=sharing

C.16 05.2017

https://drive.google.com/file/d/19qbHqW7SOupG7IaiacPgV1eYKZ1TC6vQ/view?usp=sharing

C.17 07.2017

https://drive.google.com/file/d/1KGfiwduu0Vqa1NHwr0MLFWIEzlcy-sqz/view?usp=sharing

C.18 08.2017

https://drive.google.com/file/d/1AN5XwgiQO0rwnYyj7gtj93kebawUl6Eq/view?usp=sharing

C.19 09.2017

https://drive.google.com/file/d/1AQTFOTa-lhIi92xpa1HrXjR2YUKZnYgk/view?usp=sharing

C.20 11.2017

https://drive.google.com/file/d/1bw0_N4CRe-AFuko9dFna3Lue8mW-nV9A/view?usp=sharing

C.21 12.2017

https://drive.google.com/file/d/1Gt2crt7zIE-Q3yZyv8qg9rgFISlY5rDN/view?usp=sharing

C.22 01.2018

https://drive.google.com/file/d/1q2j5OGUOaw_9FEOq3-JzJxgVDZmu2IWU/view?usp=sharing

C.23 02.2018

https://drive.google.com/file/d/15E_Ij8wyYlU4Km9hlKNKmBotjZZIKo7A/view?usp=sharing

C.24 03.2018

https://drive.google.com/file/d/174YkGd9j1z_9EzDSXRj-9FaFpXr_IwnD/view?usp=sharing

C.25 04.2018

https://drive.google.com/file/d/11znjLGyYM6VouyP9S8mTxE6TqkqePd9_/view?usp=sharing