In [1]:

```
!pip install dfply
import numpy as np
import pandas as pd
import scipy.stats as stats
import matplotlib.pyplot as plt
import math
import seaborn as sns
from dfply import *
```

```
Collecting dfply
  Downloading https://files.pythonhosted.org/package
s/53/91/18ab48c64661252dadff685f8ddbc6f456302923918f
488714ee2345d49b/dfply-0.3.3-py3-none-any.whl (612kB
)
     |████████████████████████████████| 614kB 4.8MB/
s
Requirement already satisfied: pandas in /usr/local/
lib/python3.6/dist-packages (from dfply) (1.0.3)
Requirement already satisfied: numpy in /usr/local/l
ib/python3.6/dist-packages (from dfply) (1.18.3)
Requirement already satisfied: python-dateutil>=2.6.
1 in /usr/local/lib/python3.6/dist-packages (from pa
ndas->dfply) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in /usr/
local/lib/python3.6/dist-packages (from pandas->dfpl
y) (2018.9)
Requirement already satisfied: six>=1.5 in /usr/loca
l/lib/python3.6/dist-packages (from python-dateutil>
=2.6.1->pandas->dfply) (1.12.0)
Installing collected packages: dfply
Successfully installed dfply-0.3.3

/usr/local/lib/python3.6/dist-packages/statsmodels/t
ools/_testing.py:19: FutureWarning: pandas.util.test
ing is deprecated. Use the functions in the public A
PI at pandas.testing instead.
  import pandas.util.testing as tm
```

In [0]:

```
# filter warnings on depreciation etc.
import warnings
warnings.filterwarnings("ignore")
```

In [0]:

```
#PROJECT OBJECTIVE 1
#Data was obtained from stats canada. Citation to be added later
#Prior to uploading into python environment, the csv file needed
to be cleaned to be in a format that pandas could read
```

In [0]:

```
# code to read csv file into Colaboratory:
!pip install -U -q PyDrive
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials
# Authenticate and create the PyDrive client.
auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)
```

In [0]:

```
# load data set
link = 'https://drive.google.com/open?id=13BLAQdWtZgQ_gHL0UpDiiE
ji1-1pVOz6' # The shareable link
fluff, id = link.split('=')
# loading bodies dataset
downloaded = drive.CreateFile({'id':id})
downloaded.GetContentFile('Filename.csv')
df1 = pd.read_csv('Filename.csv')
# Dataset is now stored in a Pandas Dataframe
```

In [5]:

```
df1.head()
```

Out[5]:

	geography	status_in_canada	reference_period	number
0	Canada	All	1992 / 1993	885,645
1	Canada	All	1993 / 1994	874,605
2	Canada	All	1994 / 1995	858,972
3	Canada	All	1995 / 1996	846,408
4	Canada	All	1996 / 1997	829,767

In [6]:

```
#Convert numbers under number column to integers and remove the
comma
df1['number'] = [int(i.replace(',', '')) for i in df1['number']]

#Removing the spacing in the reference_period
df1['reference_period'] = [str(i.replace(' / ', '/')) for i in d
f1['reference_period']]

df1
```

Out[6]:

	geography	status_in_canada	reference_period	number
0	Canada	All	1992/1993	885645
1	Canada	All	1993/1994	874605
2	Canada	All	1994/1995	858972
3	Canada	All	1995/1996	846408
4	Canada	All	1996/1997	829767
...	...	...	...	...
931	Territories	international	2013/2014	0
932	Territories	international	2014/2015	0
933	Territories	international	2015/2016	0
934	Territories	international	2016/2017	0
935	Territories	international	2017/2018	0

936 rows × 4 columns

In [0]:

```
#The data shows the number of international, domestic and all (i
ncludes unspecific students)
#at universities across Canada between 1992/1993 and 2017/18.
#Data for 2018/2019 was unavailable.
```

In [8]:

```
#DATA EXPLORATION

#Looking at the distinct values under georgraphy:
df1.geography.unique().tolist()
```

Out[8]:

```
['Canada',
 'Newfoundland and Labrador',
 'Prince Edward Island',
 'Nova Scotia',
 'New Brunswick',
 'Quebec',
 'Ontario',
 'Manitoba',
 'Saskatchewan',
 'Alberta',
 'British Columbia',
 'Territories']
```

In [9]:

```
#Canada

#To begin, we want to look at growth across all of Canada, begin
ning at 1992.

#Creating a subset of df1 for Canada (Geography == Canada)

Canada = df1.loc[df1['geography'] == 'Canada']
Canada
```

Out[9]:

	geography	status_in_canada	reference_period	number
0	Canada	All	1992/1993	885645
1	Canada	All	1993/1994	874605
2	Canada	All	1994/1995	858972
3	Canada	All	1995/1996	846408
4	Canada	All	1996/1997	829767
...	...	...	...	...
73	Canada	international	2013/2014	144351
74	Canada	international	2014/2015	159405
75	Canada	international	2015/2016	168591
76	Canada	international	2016/2017	179796
77	Canada	international	2017/2018	196563

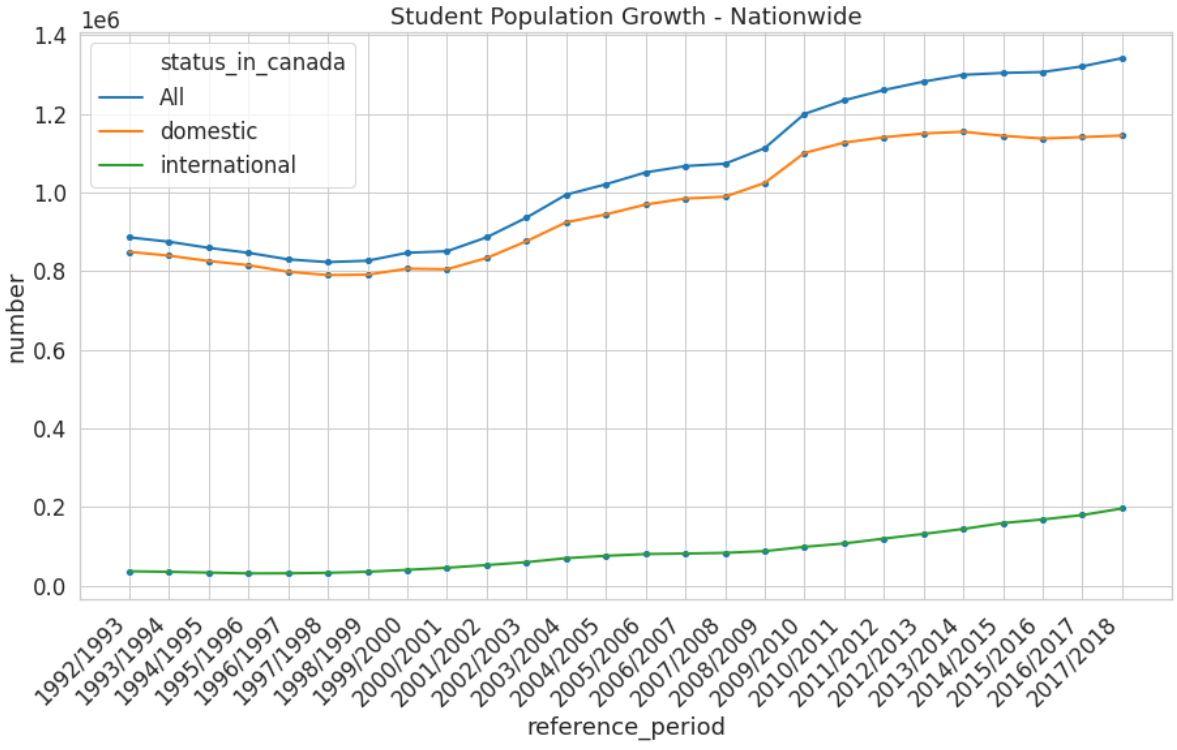
78 rows × 4 columns

In [10]:

```
sns.set_context("notebook", font_scale=1.5, rc={"lines.linewidth
": 2.0})
sns.set_style("whitegrid")

plt.figure(figsize=(15, 8))

plt.title("Student Population Growth - Nationwide")
plt.xticks(rotation=45, horizontalalignment='right')


sns.lineplot("reference_period", "number", data=Canada, hue='sta
tus_in_canada')
sns.scatterplot("reference_period", "number", data=Canada)

#Can see growth becoming more significant beginning at 2000.
#The number of domestic students remains significantly higher th
an that of international students
```

Out[10]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9e7f74
a320>
```

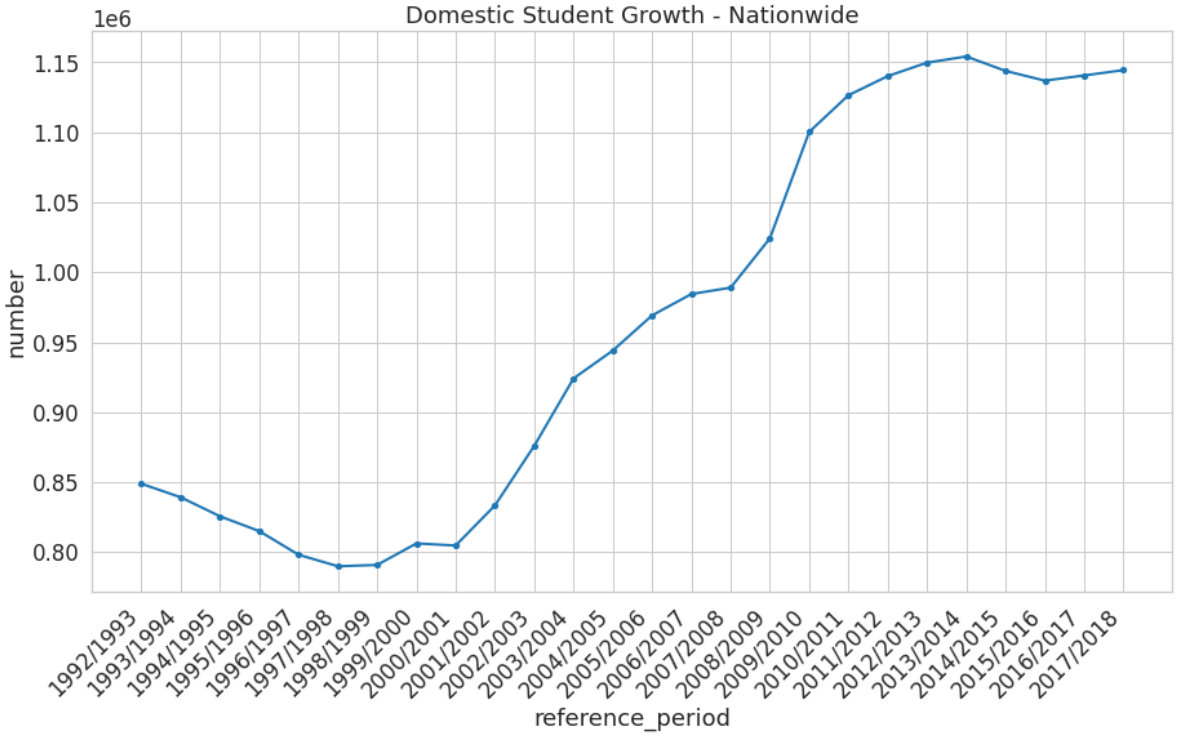


In [11]:

```
#Isolating Domestic and International - closer look

Can_domestic = Canada >> mask(X.status_in_canada == 'domestic')
Can_international = Canada >> mask(X.status_in_canada == 'intern
ational')

#Plotting Domestic - Nationwide
sns.set_context("notebook", font_scale=1.5, rc={"lines.linewidth
": 2.0})
sns.set_style("whitegrid")

plt.figure(figsize=(15, 8))

plt.title("Domestic Student Growth - Nationwide")
plt.xticks(rotation=45, horizontalalignment='right')


sns.scatterplot("reference_period", "number", data=Can_domestic)
sns.lineplot("reference_period", "number", data=Can_domestic)
```

Out[11]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9e7f7e
ce10>
```

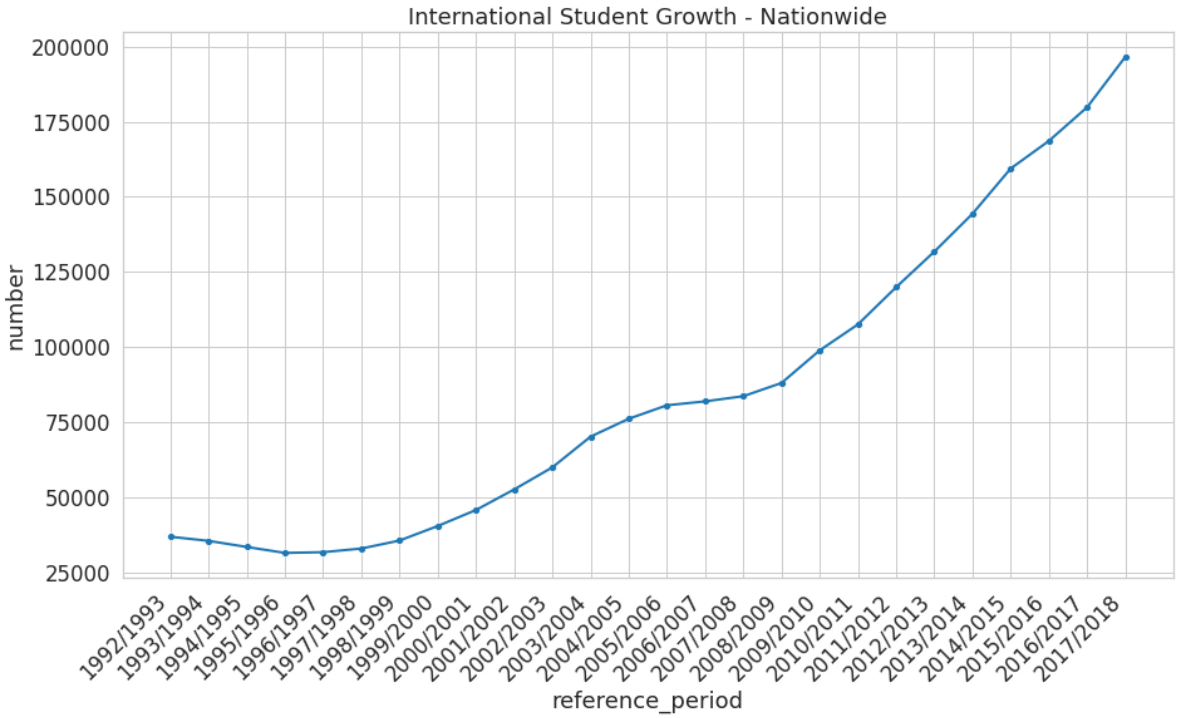


In [12]:

```
#Plotting International - Nationwide
sns.set_context("notebook", font_scale=1.5, rc={"lines.linewidth
": 2.0})
sns.set_style("whitegrid")

plt.figure(figsize=(15, 8))

plt.title("International Student Growth - Nationwide")
plt.xticks(rotation=45, horizontalalignment='right')

sns.scatterplot("reference_period", "number", data=Can_internati
onal)
sns.lineplot("reference_period", "number", data=Can_internationa
l)
```

Out[12]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9e7f6a
55f8>
```



In [13]:

```
#PROVINCES

#Looking at growth by Province (1992 to 2018)
Provinces = df1.loc[df1['geography'] != 'Canada']
Provinces
```

Out[13]:

	geography	status_in_canada	reference_period	number
78	Newfoundland and Labrador	All	1992/1993	17856
79	Newfoundland and Labrador	All	1993/1994	17397
80	Newfoundland and Labrador	All	1994/1995	17169
81	Newfoundland and Labrador	All	1995/1996	16215
82	Newfoundland and Labrador	All	1996/1997	16053
...	...	...	...	...
931	Territories	international	2013/2014	0
932	Territories	international	2014/2015	0
933	Territories	international	2015/2016	0
934	Territories	international	2016/2017	0
935	Territories	international	2017/2018	0

858 rows × 4 columns

In [0]:

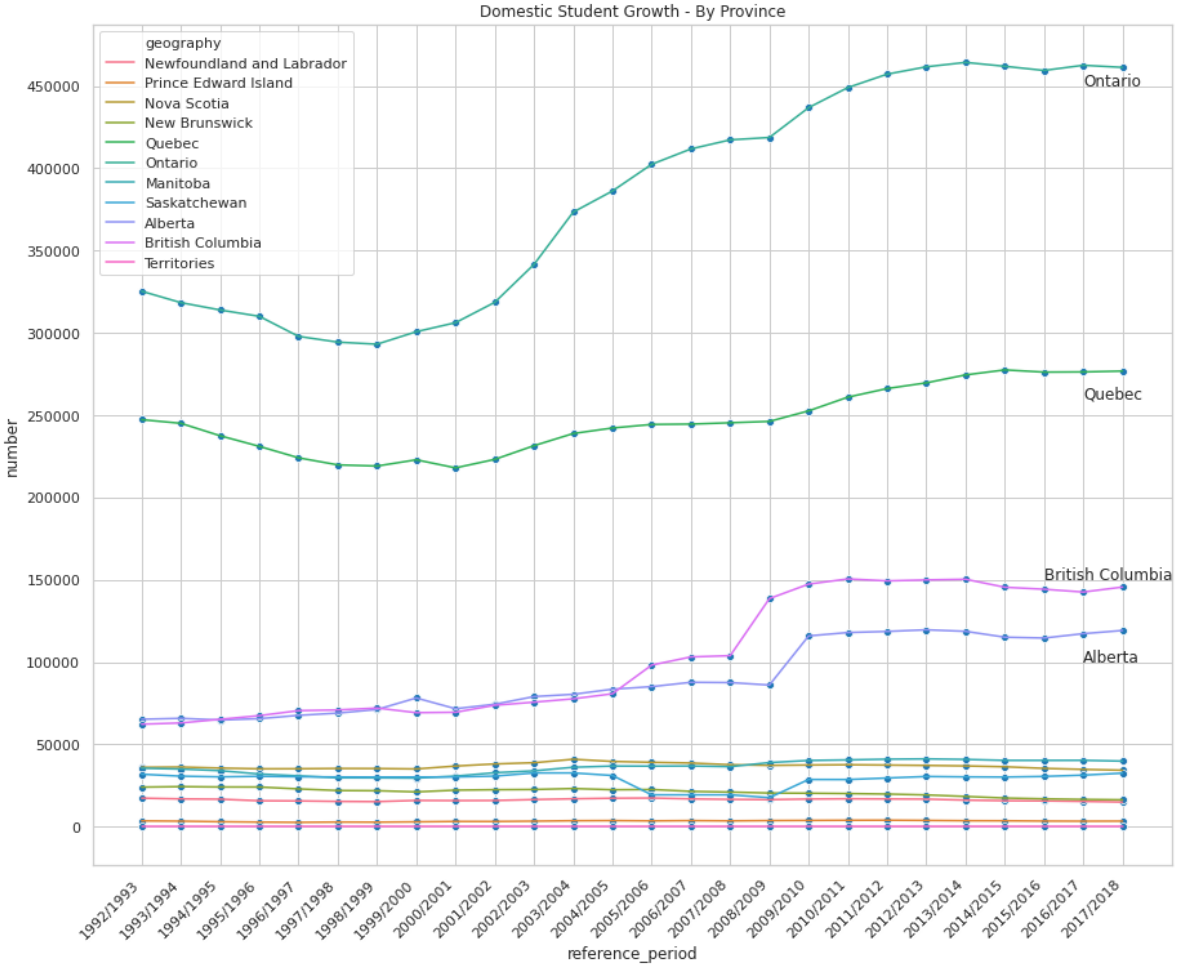
```
#Isolating domestic and international
Prov_domestic = Provinces >> mask(X.status_in_canada == 'domesti
c')
Prov_international = Provinces >> mask(X.status_in_canada == 'in
ternational')
```

In [15]:

```
#Comparison by province - domestic
sns.set_context("notebook", font_scale=1.0, rc={"lines.linewidth
": 1.5})
sns.set_style("whitegrid")
plt.figure(figsize=(15, 12))

plt.title("Domestic Student Growth - By Province")
plt.xticks(rotation=45, horizontalalignment='right')
plt.yticks([0, 50000, 100000, 150000, 200000, 250000, 300000, 35
0000, 400000 , 450000])

sns.lineplot("reference_period", "number", data=Prov_domestic, h
ue='geography')
sns.scatterplot("reference_period", "number", data=Prov_domestic
)

plt.text(24,450000, "Ontario")
plt.text(24,260000, "Quebec")
plt.text(23,150000, "British Columbia")
plt.text(24,100000, "Alberta")

#The graph below shows that the domestic student growth seen in
Canada has been driven by Ontario, Quebec, Alberta, and British
Columbia
#While growth in all other provinces has remained steady
#Will therefore focus attention on Ontario, Duebed, Alberta and
British Columbia moving forward
```

Out[15]:

```
Text(24, 100000, 'Alberta')
```

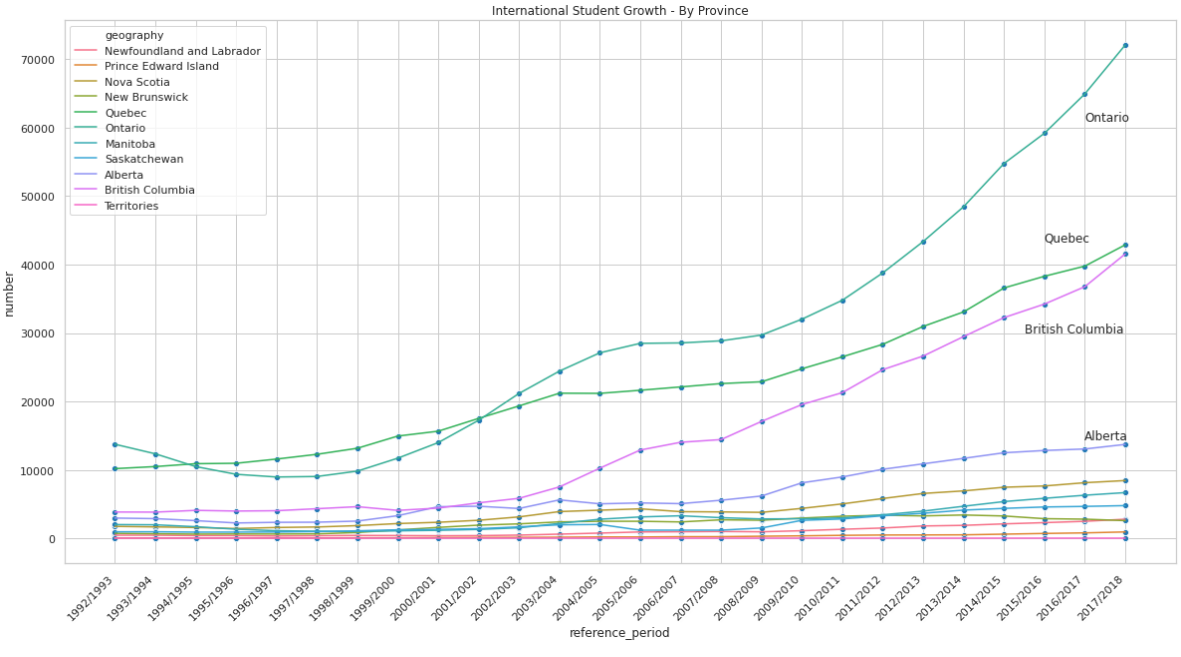


In [16]:

```
#Comparison by province - international
sns.set_context("notebook", font_scale=1.0, rc={"lines.linewidth
": 1.5})
sns.set_style("whitegrid")
plt.figure(figsize=(20, 10))

plt.title("International Student Growth - By Province")
plt.xticks(rotation=45, horizontalalignment='right')


sns.lineplot("reference_period", "number", data=Prov_internation
al, hue='geography')
sns.scatterplot("reference_period", "number", data=Prov_internat
ional)

plt.text(24,61000, "Ontario")
plt.text(23,43400, "Quebec")
plt.text(22.5,30000, "British Columbia")
plt.text(24,14500, "Alberta")

#Just like with Domestic students, Ontario, Quebec, BC and Alber
ta are the key drivers of growth.
#Will therefore focus on these provinces moving forward
```

Out[16]:

```
Text(24, 14500, 'Alberta')
```



In [0]:

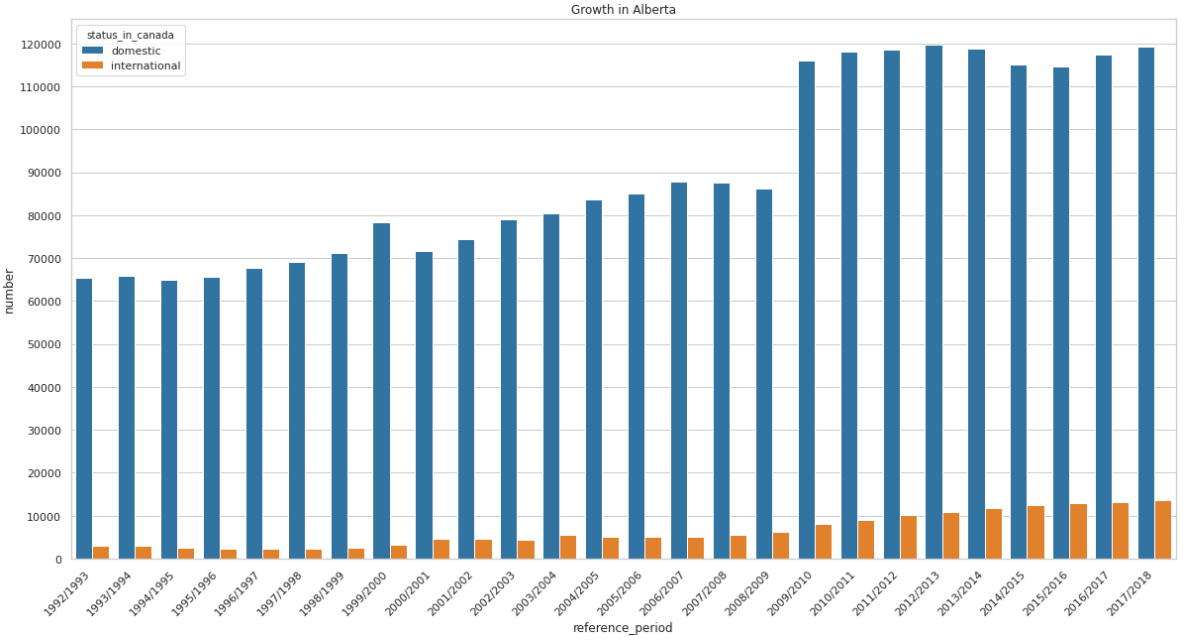
```
#Isolating Alberta, British Columbia, Ontario and Quebec
#Alberta
AB = df1.loc[df1['geography'] == 'Alberta']
AB1 = AB >> mask(X.status_in_canada != "All")
#British Columbia
BC = df1.loc[df1['geography'] == 'British Columbia']
BC1 = BC >> mask(X.status_in_canada != "All")
#Ontario
ON = df1.loc[df1['geography'] == 'Ontario']
ON1 = ON >> mask(X.status_in_canada != "All")
#Quebec
QC = df1.loc[df1['geography'] == 'Quebec']
QC1 = QC >> mask(X.status_in_canada != "All")
```

In [18]:

```
#Alberta

sns.set_context("notebook", font_scale=1.0, rc={"lines.linewidth
": 1.5})
sns.set_style("whitegrid")
plt.figure(figsize=(20, 10))

plt.title("Growth in Alberta")
plt.xticks(rotation=45, horizontalalignment='right')
plt.yticks([0, 10000, 20000, 30000, 40000, 50000, 60000, 70000,
80000, 90000, 100000, 110000, 120000, 130000, 140000])

sns.barplot("reference_period", "number", data=AB1, hue='status_
in_canada')

#A sharp growth in domestic student occured in the 2009/2010
#Can see a noticeable growth increase of international students
starting roughly the same year
```

Out[18]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9e7b03
be10>
```

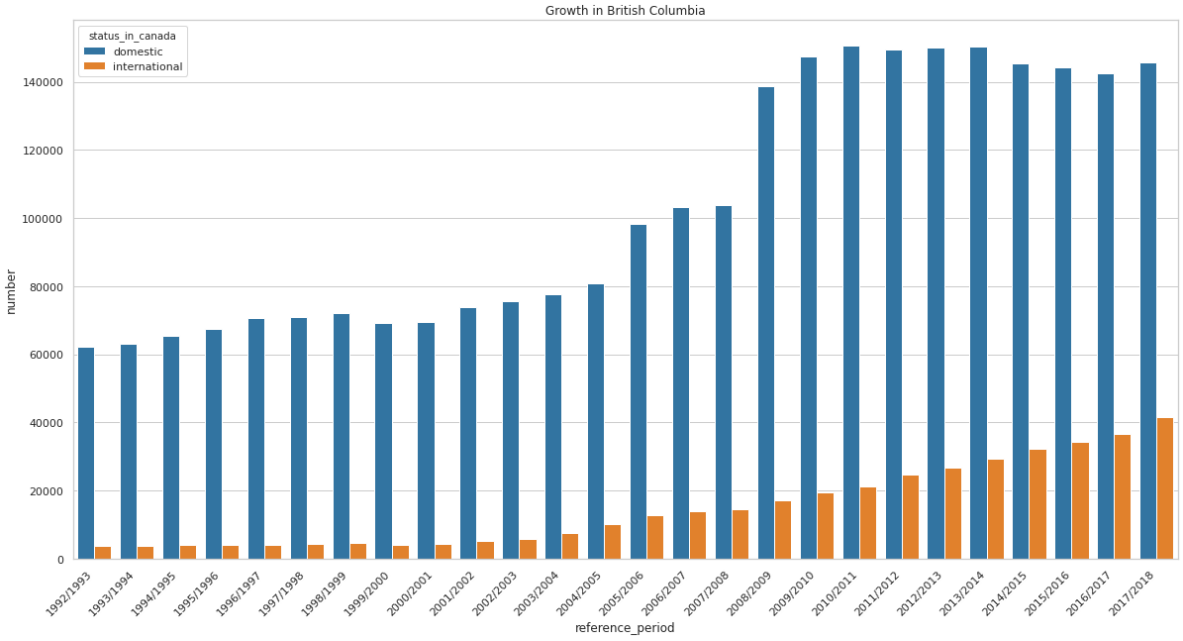


In [19]:

```
#British Columbia

sns.set_context("notebook", font_scale=1.0, rc={"lines.linewidth
": 1.5})
sns.set_style("whitegrid")
plt.figure(figsize=(20, 10))

plt.title("Growth in British Columbia")
plt.xticks(rotation=45, horizontalalignment='right')
plt.yticks([0, 20000, 40000, 60000, 80000, 100000, 120000, 14000
0, 160000, 180000])

sns.barplot("reference_period", "number", data=BC1, hue='status_
in_canada')
```

Out[19]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9e7f63
5898>
```

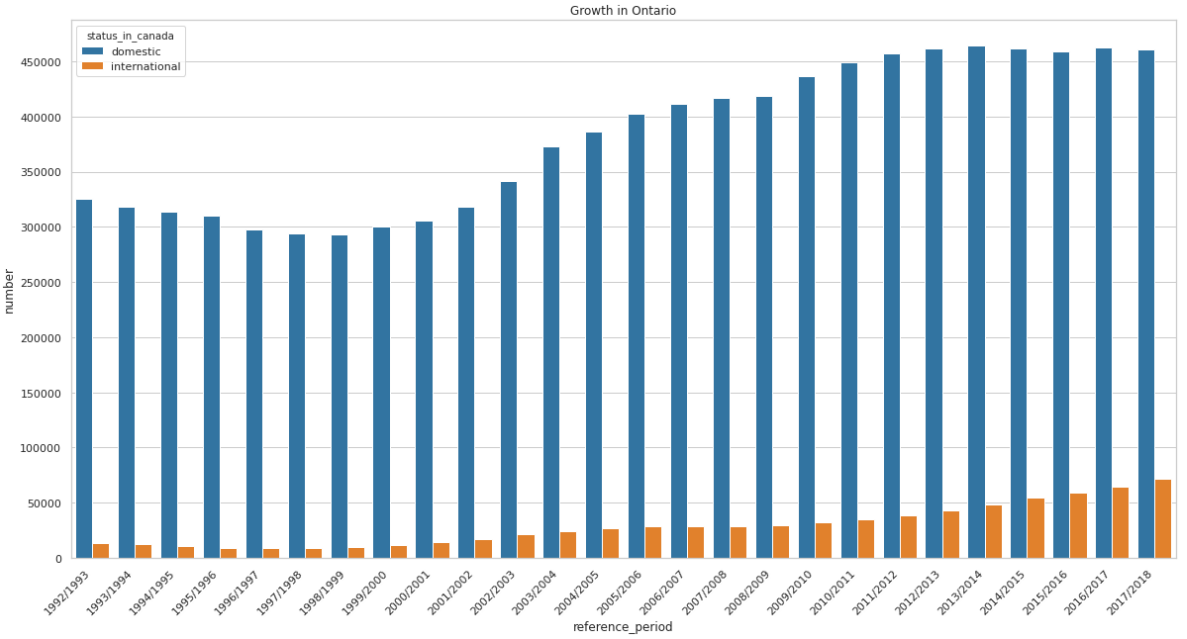


In [20]:

```
#Ontario
sns.set_context("notebook", font_scale=1.0, rc={"lines.linewidth
": 1.5})
sns.set_style("whitegrid")
plt.figure(figsize=(20, 10))

plt.title("Growth in Ontario")
plt.xticks(rotation=45, horizontalalignment='right')
plt.yticks([0, 50000, 100000, 150000, 200000, 250000, 300000, 35
0000, 400000, 450000, 500000, 550000])

sns.barplot("reference_period", "number", data=ON1, hue='status_
in_canada')
```

Out[20]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9e7cdc
ec50>
```

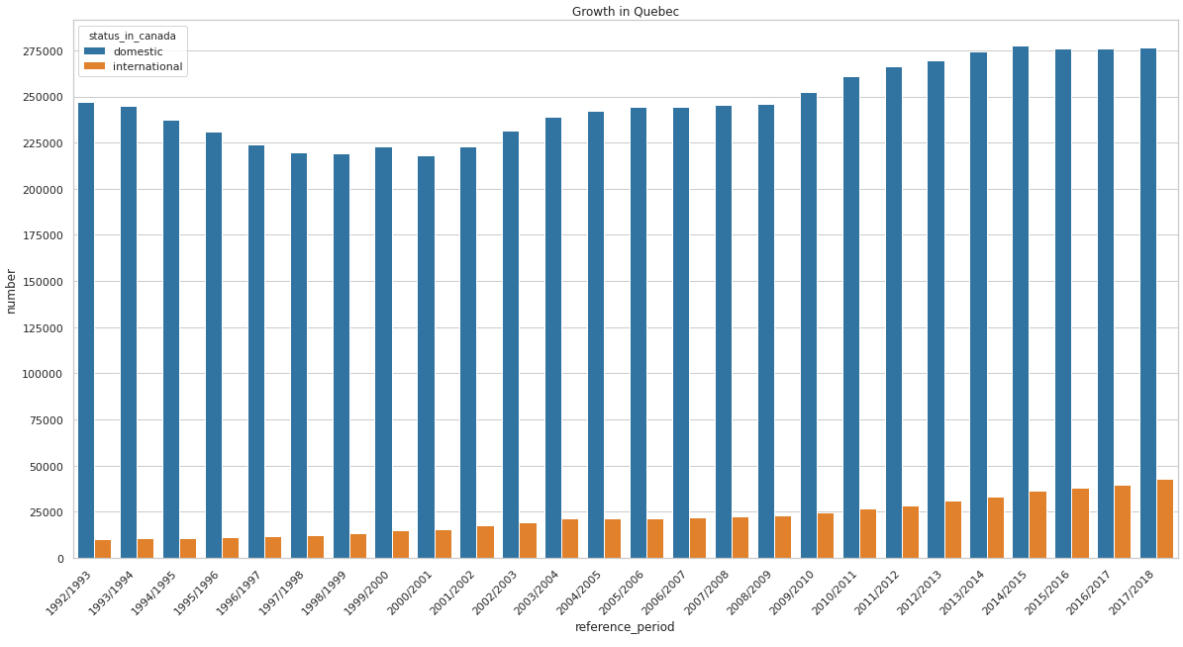


In [21]:

```
#Quebec
sns.set_context("notebook", font_scale=1.0, rc={"lines.linewidth
": 1.5})
sns.set_style("whitegrid")
plt.figure(figsize=(20, 10))

plt.title("Growth in Quebec")
plt.xticks(rotation=45, horizontalalignment='right')
plt.yticks([0, 25000, 50000, 75000, 100000, 125000, 150000, 1750
00, 200000, 225000, 250000, 275000, 300000, 325000])

sns.barplot("reference_period", "number", data=QC1, hue='status_
in_canada')
```

Out[21]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9e7a8e
c4e0>
```



In [0]:

```
########  HERE SHOULD PLOT THE RATIO OF INTERNATION:DOMESTIC TO
SHOW CHANGE OVER TIME  ##############################
```

In [0]:

```
####  SINCE THE CURRENT SCHOOL YEAR DATA DOES NOT SEPARATE THE T
OTAL ENROLLMENT BY INTERNATIONAL AND DOMESTIC
####  Determining the predicted proportion of all students in 20
19/2020 that are likely to be international
#### Using tables AB, BC, ON, and QC
AB_1 = AB >> arrange(X.reference_period)
BC_1 = BC >> arrange(X.reference_period)
ON_1 = ON >> arrange(X.reference_period)
QC_1 = QC >> arrange(X.reference_period)
```

In [0]:

```
#filter include data from 2008/2009 onward - noticable growth st
arting in that school year

year_range2 = ['2008/2009','2009/2010','2010/2011','2011/2012','
2012/2013','2013/2014','2014/2015','2015/2016','2016/2017','2017
/2018']
AB_1 = AB_1 >> mask(X.reference_period.isin(year_range2))
BC_1 = BC_1 >> mask(X.reference_period.isin(year_range2))
ON_1 = ON_1 >> mask(X.reference_period.isin(year_range2))
QC_1 = QC_1 >> mask(X.reference_period.isin(year_range2))
```

In [24]:

```
# create a column to make a continuous variable for year - conve
rting reference period to a float

#for AB
start_years = []
for idx, row in AB_1.iterrows():
    start_years.append(float(row['reference_period'].split('/')[
0]))

AB_1['start_year'] = start_years

AB_1.head()

#for BC
start_years = []
for idx, row in BC_1.iterrows():
    start_years.append(float(row['reference_period'].split('/')[
0]))

BC_1['start_year'] = start_years

BC_1.head()

#for ON
start_years = []
for idx, row in ON_1.iterrows():
    start_years.append(float(row['reference_period'].split('/')[
0]))

ON_1['start_year'] = start_years

ON_1.head()

#for QC
start_years = []
for idx, row in QC_1.iterrows():
    start_years.append(float(row['reference_period'].split('/')[
0]))

QC_1['start_year'] = start_years

QC_1.head()
```

Out[24]:

	geography	status_in_canada	reference_period	number	start_year
432	Quebec	domestic	2008/2009	246213	2008.0
406	Quebec	All	2008/2009	269097	2008.0
458	Quebec	international	2008/2009	22884	2008.0
433	Quebec	domestic	2009/2010	252615	2009.0
459	Quebec	international	2009/2010	24780	2009.0

In [25]:

```
# Creating a table with percentages
#for AB
AB_PC = pd.DataFrame(columns = ['geography', 'status_in_canada',
'reference_period', 'proportion', 'start_year'])

idx = 0
for yr in year_range2:
    year_dat = AB_1[AB_1["reference_period"] == yr]  #selects da
ta for the matching year
    count_all = float(year_dat[year_dat["status_in_canada"] == '
All'][['number']].values[0]) #selects the all number

    # compute domestic percentages
    dom_ratio = float(year_dat[year_dat["status_in_canada"] == '
domestic'][['number']].values[0]) / count_all
    AB_PC.loc[idx] = ['Alberta', 'domestic', yr, dom_ratio, floa
t(yr.split('/')[0])]

    idx += 1

for yr in year_range2:
    year_dat = AB_1[AB_1["reference_period"] == yr]
    count_all = float(year_dat[year_dat["status_in_canada"] == '
All'][['number']].values[0])

    # compute international percentages
    int_ratio = float(year_dat[year_dat["status_in_canada"] == '
international'][['number']].values[0]) / count_all
    AB_PC.loc[idx] = ['Alberta', 'international', yr, int_ratio,
float(yr.split('/')[0])]

    idx += 1

AB_PC >> arrange(X.reference_period)
```

Out[25]:

	geography	status_in_canada	reference_period	proportion	start_yea
0	Alberta	domestic	2008/2009	0.929606	2008.(
10	Alberta	international	2008/2009	0.066896	2008.(
1	Alberta	domestic	2009/2010	0.932571	2009.(
11	Alberta	international	2009/2010	0.065354	2009.(
2	Alberta	domestic	2010/2011	0.928727	2010.(
12	Alberta	international	2010/2011	0.070707	2010.(
3	Alberta	domestic	2011/2012	0.920579	2011.(
13	Alberta	international	2011/2012	0.078373	2011.(
4	Alberta	domestic	2012/2013	0.915936	2012.(
14	Alberta	international	2012/2013	0.083421	2012.(
5	Alberta	domestic	2013/2014	0.909613	2013.(
15	Alberta	international	2013/2014	0.089605	2013.(
6	Alberta	domestic	2014/2015	0.901496	2014.(
16	Alberta	international	2014/2015	0.097964	2014.(
17	Alberta	international	2015/2016	0.100649	2015.(
7	Alberta	domestic	2015/2016	0.898200	2015.(
18	Alberta	international	2016/2017	0.100163	2016.(
8	Alberta	domestic	2016/2017	0.899216	2016.(
9	Alberta	domestic	2017/2018	0.896198	2017.(
19	Alberta	international	2017/2018	0.103238	2017.(

In [26]:

```
#for BC
BC_PC = pd.DataFrame(columns = ['geography', 'status_in_canada',
'reference_period', 'proportion', 'start_year'])

idx = 0
for yr in year_range2:
    year_dat = BC_1[BC_1["reference_period"] == yr]  #selects da
ta for the matching year
    count_all = float(year_dat[year_dat["status_in_canada"] == '
All'][['number']].values[0]) #selects the all number

    # compute domestic percentages
    dom_ratio = float(year_dat[year_dat["status_in_canada"] == '
domestic'][['number']].values[0]) / count_all
    BC_PC.loc[idx] = ['British Columbia', 'domestic', yr, dom_ra
tio, float(yr.split('/')[0])]

    idx += 1

for yr in year_range2:
    year_dat = BC_1[BC_1["reference_period"] == yr]
    count_all = float(year_dat[year_dat["status_in_canada"] == '
All'][['number']].values[0])

    # compute international percentages
    int_ratio = float(year_dat[year_dat["status_in_canada"] == '
international'][['number']].values[0]) / count_all
    BC_PC.loc[idx] = ['British Columbia', 'international', yr, i
nt_ratio, float(yr.split('/')[0])]

    idx += 1

BC_PC >> arrange(X.reference_period)
```

Out[26]:

	geography	status_in_canada	reference_period	proportion	start_yea
0	British Columbia	domestic	2008/2009	0.890223	2008.(
	British				

10	Columbia	international	2008/2009	0.109797	2008.(
1	British Columbia	domestic	2009/2010	0.882841	2009.(
11	British Columbia	international	2009/2010	0.117159	2009.(
2	British Columbia	domestic	2010/2011	0.876065	2010.(
12	British Columbia	international	2010/2011	0.123953	2010.(
3	British Columbia	domestic	2011/2012	0.858305	2011.(
13	British Columbia	international	2011/2012	0.141677	2011.(
4	British Columbia	domestic	2012/2013	0.849118	2012.(
14	British Columbia	international	2012/2013	0.150899	2012.(
5	British Columbia	domestic	2013/2014	0.836033	2013.(
15	British Columbia	international	2013/2014	0.163967	2013.(
6	British Columbia	domestic	2014/2015	0.818545	2014.(
16	British Columbia	international	2014/2015	0.181471	2014.(
17	British Columbia	international	2015/2016	0.191778	2015.(
7	British Columbia	domestic	2015/2016	0.808222	2015.(
18	British Columbia	international	2016/2017	0.205025	2016.(
8	British Columbia	domestic	2016/2017	0.794992	2016.(
9	British Columbia	domestic	2017/2018	0.777892	2017.(
19	British Columbia	international	2017/2018	0.222028	2017.(

In [27]:

```
#for ON
ON_PC = pd.DataFrame(columns = ['geography', 'status_in_canada',
'reference_period', 'proportion', 'start_year'])

idx = 0
for yr in year_range2:
    year_dat = ON_1[ON_1["reference_period"] == yr]  #selects da
ta for the matching year
    count_all = float(year_dat[year_dat["status_in_canada"] == '
All'][['number']].values[0]) #selects the all number

    # compute domestic percentages
    dom_ratio = float(year_dat[year_dat["status_in_canada"] == '
domestic'][['number']].values[0]) / count_all
    ON_PC.loc[idx] = ['Ontario', 'domestic', yr, dom_ratio, floa
t(yr.split('/')[0])]

    idx += 1

for yr in year_range2:
    year_dat = ON_1[ON_1["reference_period"] == yr]
    count_all = float(year_dat[year_dat["status_in_canada"] == '
All'][['number']].values[0])

    # compute international percentages
    int_ratio = float(year_dat[year_dat["status_in_canada"] == '
international'][['number']].values[0]) / count_all
    ON_PC.loc[idx] = ['Ontario', 'international', yr, int_ratio,
float(yr.split('/')[0])]

    idx += 1

ON_PC >> arrange(X.reference_period)
```

Out[27]:

	geography	status_in_canada	reference_period	proportion	start_yea
0	Ontario	domestic	2008/2009	0.933740	2008.(
10	Ontario	international	2008/2009	0.066260	2008.(
1	Ontario	domestic	2009/2010	0.931686	2009.(
11	Ontario	international	2009/2010	0.068314	2009.(
2	Ontario	domestic	2010/2011	0.928079	2010.(
12	Ontario	international	2010/2011	0.071915	2010.(
3	Ontario	domestic	2011/2012	0.921780	2011.(
13	Ontario	international	2011/2012	0.078214	2011.(
4	Ontario	domestic	2012/2013	0.914106	2012.(
14	Ontario	international	2012/2013	0.085888	2012.(
5	Ontario	domestic	2013/2014	0.905491	2013.(
15	Ontario	international	2013/2014	0.094503	2013.(
6	Ontario	domestic	2014/2015	0.893993	2014.(
16	Ontario	international	2014/2015	0.106007	2014.(
17	Ontario	international	2015/2016	0.114130	2015.(
7	Ontario	domestic	2015/2016	0.885870	2015.(
18	Ontario	international	2016/2017	0.123102	2016.(
8	Ontario	domestic	2016/2017	0.876892	2016.(
9	Ontario	domestic	2017/2018	0.864849	2017.(
19	Ontario	international	2017/2018	0.135151	2017.(

In [28]:

```
#for AB
QC_PC = pd.DataFrame(columns = ['geography', 'status_in_canada',
'reference_period', 'proportion', 'start_year'])

idx = 0
for yr in year_range2:
    year_dat = QC_1[QC_1["reference_period"] == yr]  #selects da
ta for the matching year
    count_all = float(year_dat[year_dat["status_in_canada"] == '
All'][['number']].values[0]) #selects the all number

    # compute domestic percentages
    dom_ratio = float(year_dat[year_dat["status_in_canada"] == '
domestic'][['number']].values[0]) / count_all
    QC_PC.loc[idx] = ['Quebec', 'domestic', yr, dom_ratio, float
(yr.split('/')[0])]

    idx += 1

for yr in year_range2:
    year_dat = QC_1[QC_1["reference_period"] == yr]
    count_all = float(year_dat[year_dat["status_in_canada"] == '
All'][['number']].values[0])

    # compute international percentages
    int_ratio = float(year_dat[year_dat["status_in_canada"] == '
international'][['number']].values[0]) / count_all
    QC_PC.loc[idx] = ['Quebec', 'international', yr, int_ratio,
float(yr.split('/')[0])]

    idx += 1

QC_PC >> arrange(X.reference_period)
```

Out[28]:

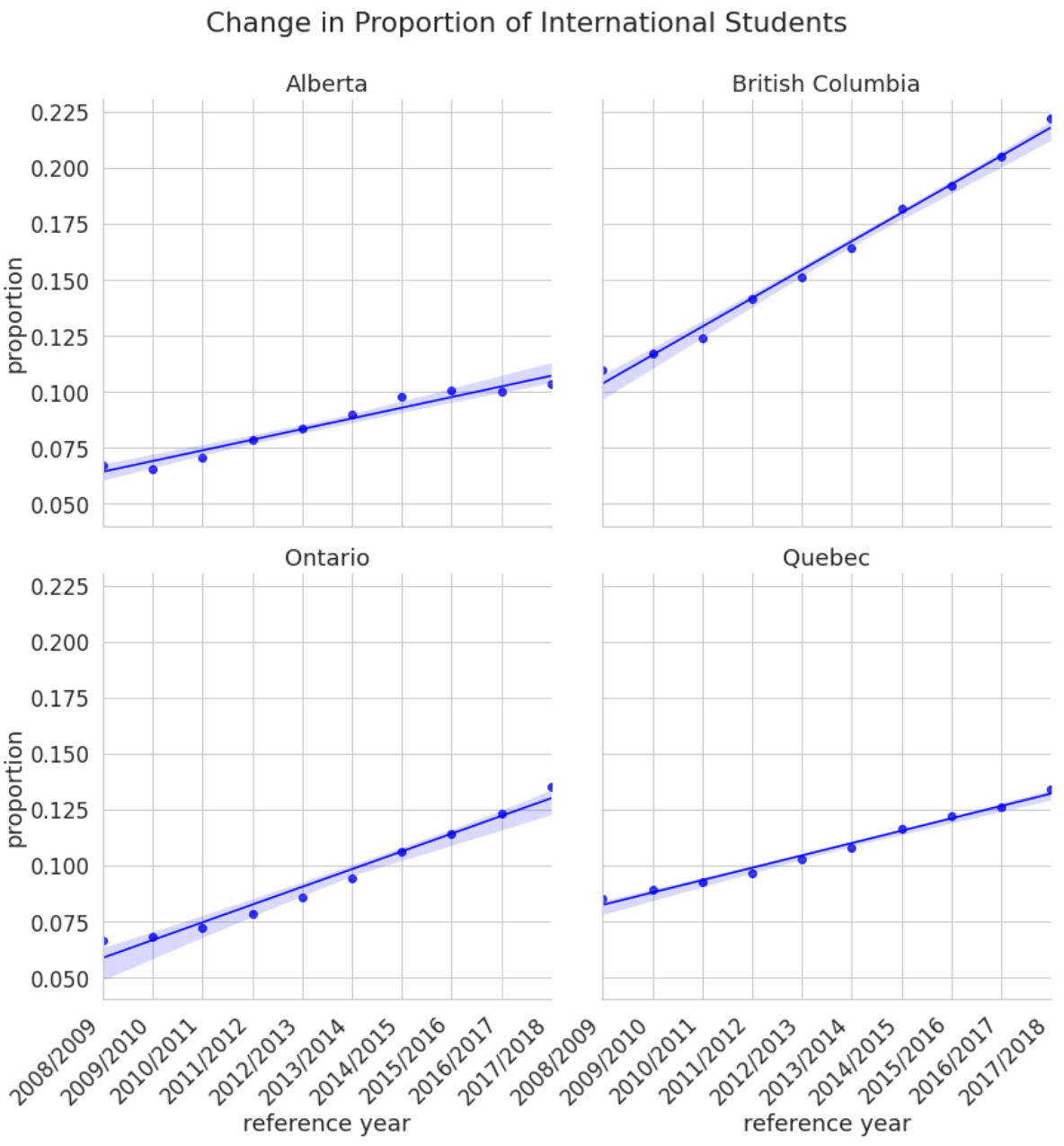
	geography	status_in_canada	reference_period	proportion	start_yea
0	Quebec	domestic	2008/2009	0.914960	2008.(
10	Quebec	international	2008/2009	0.085040	2008.(
1	Quebec	domestic	2009/2010	0.910659	2009.(
11	Quebec	international	2009/2010	0.089330	2009.(
2	Quebec	domestic	2010/2011	0.907707	2010.(
12	Quebec	international	2010/2011	0.092283	2010.(
3	Quebec	domestic	2011/2012	0.903733	2011.(
13	Quebec	international	2011/2012	0.096267	2011.(
4	Quebec	domestic	2012/2013	0.896980	2012.(
14	Quebec	international	2012/2013	0.103020	2012.(
5	Quebec	domestic	2013/2014	0.892407	2013.(
15	Quebec	international	2013/2014	0.107593	2013.(
6	Quebec	domestic	2014/2015	0.883507	2014.(
16	Quebec	international	2014/2015	0.116493	2014.(
17	Quebec	international	2015/2016	0.121783	2015.(
7	Quebec	domestic	2015/2016	0.878217	2015.(
18	Quebec	international	2016/2017	0.125865	2016.(
8	Quebec	domestic	2016/2017	0.874126	2016.(
9	Quebec	domestic	2017/2018	0.865828	2017.(
19	Quebec	international	2017/2018	0.134163	2017.(

In [29]:

```
#Performing a linear regression on the proportions in order to p
redict/estimate the
#proportion of students for 2019/2020 that are international

#filtering for internationals
AB_PC_int = AB_PC >> mask(X.status_in_canada == 'international')
BC_PC_int = BC_PC >> mask(X.status_in_canada == 'international')
ON_PC_int = ON_PC >> mask(X.status_in_canada == 'international')
QC_PC_int = QC_PC >> mask(X.status_in_canada == 'international')

#Appending
internationals = AB_PC_int.append(BC_PC_int, ignore_index = True
).append(ON_PC_int, ignore_index = True).append(QC_PC_int, ignor
e_index = True)

#plotting
sns.set_context("notebook", font_scale=1.5, rc={"lines.linewidth
": 1.0})
sns.set_style("whitegrid")

p = sns.FacetGrid(internationals, col='geography', col_wrap=2, m
argin_titles=True, height=6, aspect=1)
p = ((p.map(sns.regplot, 'start_year', 'proportion', color="blue
").set_titles("{col_name}"))
     .set_xlabels('reference year')
     .set(xticks=[2008.0, 2009.0, 2010.0, 2011.0, 2012.0, 2013.0
, 2014.0, 2015.0, 2016.0, 2017.0])
     .set_xticklabels(['2008/2009','2009/2010','2010/2011','2011
/2012','2012/2013','2013/2014','2014/2015','2015/2016','2016/201
7','2017/2018'],
                      rotation=45, horizontalalignment='right'))

plt.subplots_adjust(top=0.9)
p.fig.suptitle('Change in Proportion of International Students')
```

Out[29]:

```
Text(0.5, 0.98, 'Change in Proportion of Internation
al Students')
```



In [30]:

```
#obtaining slope and intercept information to make prediction
#Compiling estimated proportions into a table called pred_table

def get_fit(data, year):
    slope, intercept, rvalue, pvalue, stderr = stats.linregress(
data["start_year"], data["proportion"])
    return (slope * float(year) + intercept) * 100.0

pred_table = pd.DataFrame(columns=['geography', '2018', '2019'])

provinces = ['Alberta', 'British Columbia', 'Ontario', 'Quebec']
for province in provinces:

    prov_data = internationals >> mask(X.geography == province)

    pred18 = get_fit(prov_data, 2018.)
    pred19 = get_fit(prov_data, 2019.)

    row_out = pd.DataFrame([[province, pred18, pred19]], columns
=['geography', '2018', '2019'])
    pred_table = pred_table.append(row_out)

pred_table
```

Out[30]:

	geography	2018	2019
0	Alberta	11.181753	11.657763
0	British Columbia	23.066620	24.337363
0	Ontario	13.790191	14.582072
0	Quebec	13.753695	14.305574

In [0]:

```
# OBJECTIVE 2

#Looking at the amount of available housing at each univeristy i
n the 4 provinces of interest, both on and off campus.
#If the number of available beds (or units) is less that the num
ber of international students, then there is a definite gap.
#Assumption: Since we do not know how many students have applied
/been rejected from housing, we are assuming that
#all international students will be in need of housing.
#Once we can identify universities with a gap, we will look at t
he average price per bed.
```

In [0]:

```
# load data set
link = 'https://drive.google.com/open?id=1RhazTydj__YLs4Z_xFmX78
vucHhQyyRq' # The shareable link
fluff, id = link.split('=')
# loading bodies dataset
downloaded = drive.CreateFile({'id':id})
downloaded.GetContentFile('Filename.csv')
df2 = pd.read_csv('Filename.csv')
# Dataset is now stored in a Pandas Dataframe
```

In [0]:

```
df2 = df2.drop(35)  #Removing row of NaN

#Converting Total to integers
df2['Total'] = [int(i.replace(',', '')) for i in df2['Total']]
#df2
```

In [135]:

```
#Sorting by province and removing any outside of AB, ON, BC or QC
prov_string = ['Alberta', 'British Columbia', 'Ontario', 'Québec']
uni_by_prov = df2 >> arrange(X.Province) >> mask(X.Province.isin(prov_string))

#Keeping columns: University, Total, Province
uni_by_prov = uni_by_prov >> select(X.Province, X.University, X.Total)

uni_by_prov
```

Out[135]:

	Province	University	Total
30	Alberta	University of Lethbridge	8956
51	Alberta	Concordia University of Edmonton	2771
65	Alberta	The King's University	850
5	Alberta	University of Alberta	39938
10	Alberta	University of Calgary	34236
14	British Columbia	Simon Fraser University	30254
48	British Columbia	University of Northern British Columbia	3444
28	British Columbia	Thompson Rivers University	9595
55	British Columbia	Emily Carr University of Art + Design	1893
44	British Columbia	Trinity Western University	4434
33	British Columbia	Vancouver Island University	7594
1	British Columbia	The University of British Columbia	61547
17	British Columbia	University of Victoria	22134
47	Ontario	Victoria University (includes Emmanuel College)	3553
59	Ontario	Algoma University	1370
60	Ontario	Huron University College	1312

67	Ontario	Redeemer University College	789
40	Ontario	OCAD University	4820
46	Ontario	King's University College at Western University	3800
52	Ontario	Royal Military College of Canada	2585
0	Ontario	University of Toronto	92000
13	Ontario	University of Guelph	30310
2	Ontario	York University	55600
3	Ontario	Ryerson University	47350
9	Ontario	McMaster University	35040
32	Ontario	Lakehead University	8620
11	Ontario	Carleton University	31790
26	Ontario	Trent University	10880
12	Ontario	Queen's University	30500
37	Ontario	Nipissing University	5090
23	Ontario	University of Windsor	16480
21	Ontario	Brock University	19560
19	Ontario	Wilfrid Laurier University	21100
8	Québec	McGill University	36923
41	Québec	Université TÉLUQ	4570
4	Québec	Université Laval	44013
6	Québec	Concordia University	39274
7	Québec	Université du Québec à Montréal	38017
50	Québec	Bishop's University	3002
42	Québec	Université du Québec en Abitibi-Témiscamingue	4560
34	Québec	Université du Québec en Outaouais	6974
56	Québec	École nationale d'administration publique	1759
24	Québec	Université du Québec à Trois-Rivières	14203
16	Québec	Université de Sherbrooke	25161

31	Québec	École de technologie supérieure	8843

In [35]:

```
#Mutating this table to break the total into estimated proportio
n of international and domestic
#Alberta - 11.657763% for 2019
#British Columbia - 24.337363% for 2019
#Ontario - 14.582072% for 2019
#Quebec - 14.305574%

AB_int_dom = uni_by_prov >> mask(X.Province == 'Alberta') >> mut
ate(International = X.Total*0.11657763) >> mutate(Domestic = X.T
otal-X.International)
BC_int_dom = uni_by_prov >> mask(X.Province == 'British Columbia
') >> mutate(International = X.Total*0.24337363) >> mutate(Domes
tic = X.Total-X.International)
ON_int_dom = uni_by_prov >> mask(X.Province == 'Ontario') >> mut
ate(International = X.Total*0.14582072) >> mutate(Domestic = X.T
otal-X.International)
QC_int_dom = uni_by_prov >> mask(X.Province == 'Québec') >> muta
te(International = X.Total*0.14305574) >> mutate(Domestic = X.To
tal-X.International)

uni_prop = AB_int_dom.append(BC_int_dom, ignore_index = True).ap
pend(ON_int_dom, ignore_index = True).append(QC_int_dom, ignore_
index = True)
uni_prop['International'] = round(uni_prop['International'])
uni_prop['Domestic'] = round(uni_prop['Domestic'])

#Creating a column of ratio of International:Domestic
uni_prop = uni_prop >> mutate(Ratio = X.International/X.Domestic
)

uni_prop
```

Out[35]:

	Province	University	Total	International	Domestic	Ratio
0	Alberta	University of Lethbridge	8956	1044.0	7912.0	0.131951
1	Alberta	Concordia University of	2771	323.0	2448.0	0.131944

		Edmonton				
2	Alberta	The King's University	850	99.0	751.0	0.131824
3	Alberta	University of Alberta	39938	4656.0	35282.0	0.131965
4	Alberta	University of Calgary	34236	3991.0	30245.0	0.131956
5	British Columbia	Simon Fraser University	30254	7363.0	22891.0	0.321655
6	British Columbia	University of Northern British Columbia	3444	838.0	2606.0	0.321566
7	British Columbia	Thompson Rivers University	9595	2335.0	7260.0	0.321625
8	British Columbia	Emily Carr University of Art + Design	1893	461.0	1432.0	0.321927
9	British Columbia	Trinity Western University	4434	1079.0	3355.0	0.321610
10	British Columbia	Vancouver Island University	7594	1848.0	5746.0	0.321615
11	British Columbia	The University of British Columbia	61547	14979.0	46568.0	0.321659
12	British Columbia	University of Victoria	22134	5387.0	16747.0	0.321670
13	Ontario	Victoria University (includes Emmanuel College)	3553	518.0	3035.0	0.170675
14	Ontario	Algoma University	1370	200.0	1170.0	0.170940
15	Ontario	Huron University College	1312	191.0	1121.0	0.170384
16	Ontario	Redeemer University College	789	115.0	674.0	0.170623

17	Ontario	OCAD University	4820	703.0	4117.0	0.170755
18	Ontario	King's University College at Western University	3800	554.0	3246.0	0.170672
19	Ontario	Royal Military College of Canada	2585	377.0	2208.0	0.170743
20	Ontario	University of Toronto	92000	13416.0	78584.0	0.170722
21	Ontario	University of Guelph	30310	4420.0	25890.0	0.170722
22	Ontario	York University	55600	8108.0	47492.0	0.170723
23	Ontario	Ryerson University	47350	6905.0	40445.0	0.170726
24	Ontario	McMaster University	35040	5110.0	29930.0	0.170732
25	Ontario	Lakehead University	8620	1257.0	7363.0	0.170718
26	Ontario	Carleton University	31790	4636.0	27154.0	0.170730
27	Ontario	Trent University	10880	1587.0	9293.0	0.170774
28	Ontario	Queen's University	30500	4448.0	26052.0	0.170735
29	Ontario	Nipissing University	5090	742.0	4348.0	0.170653
30	Ontario	University of Windsor	16480	2403.0	14077.0	0.170704
31	Ontario	Brock University	19560	2852.0	16708.0	0.170697
32	Ontario	Wilfrid Laurier University	21100	3077.0	18023.0	0.170726
33	Québec	McGill University	36923	5282.0	31641.0	0.166935
34	Québec	Université TÉLUQ	4570	654.0	3916.0	0.167007

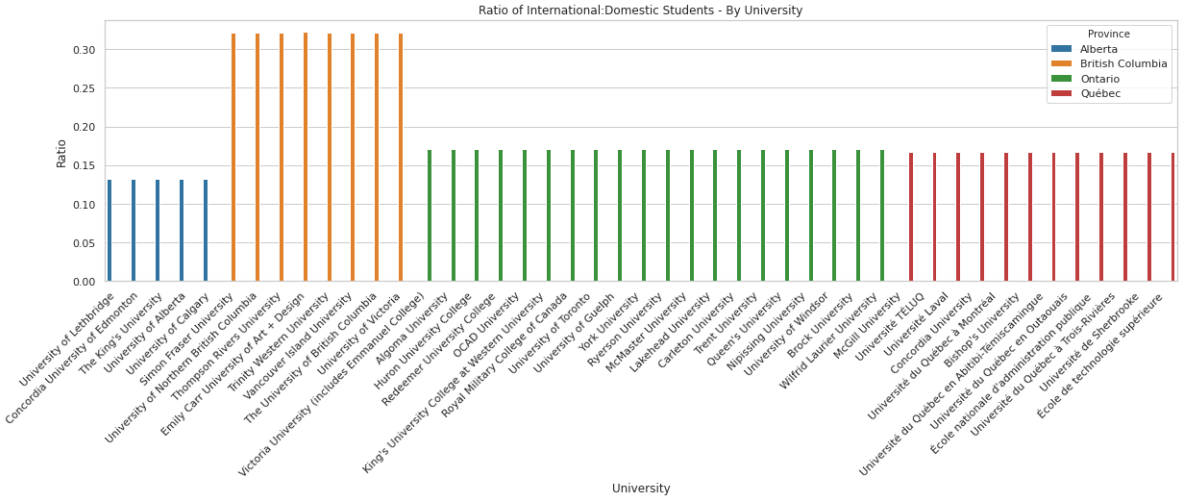
35	Québec	Université Laval	44013	6296.0	37717.0	0.166927
36	Québec	Concordia University	39274	5618.0	33656.0	0.166924
37	Québec	Université du Québec à Montréal	38017	5439.0	32578.0	0.166953
38	Québec	Bishop's University	3002	429.0	2573.0	0.166731
39	Québec	Université du Québec en Abitibi-Témiscamingue	4560	652.0	3908.0	0.166837
40	Québec	Université du Québec en Outaouais	6974	998.0	5976.0	0.167001
41	Québec	École nationale d'administration publique	1759	252.0	1507.0	0.167220
42	Québec	Université du Québec à Trois-Rivières	14203	2032.0	12171.0	0.166954
43	Québec	Université de Sherbrooke	25161	3599.0	21562.0	0.166914
44	Québec	École de technologie supérieure	8843	1265.0	7578.0	0.166931

In [36]:

```
sns.set_context("notebook", font_scale=1.0, rc={"lines.linewidth
": 1.5})
sns.set_style("whitegrid")
plt.figure(figsize=(20, 5))

plt.title("Ratio of International:Domestic Students - By Univers
ity")
plt.xticks(rotation=45, horizontalalignment='right')

sns.barplot("University", "Ratio", data=uni_prop, hue='Province'
)

#This tells us that British Columbia intakes a greater proportio
n of international students than the other provinces
```

Out[36]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9e7a55
bcf8>
```

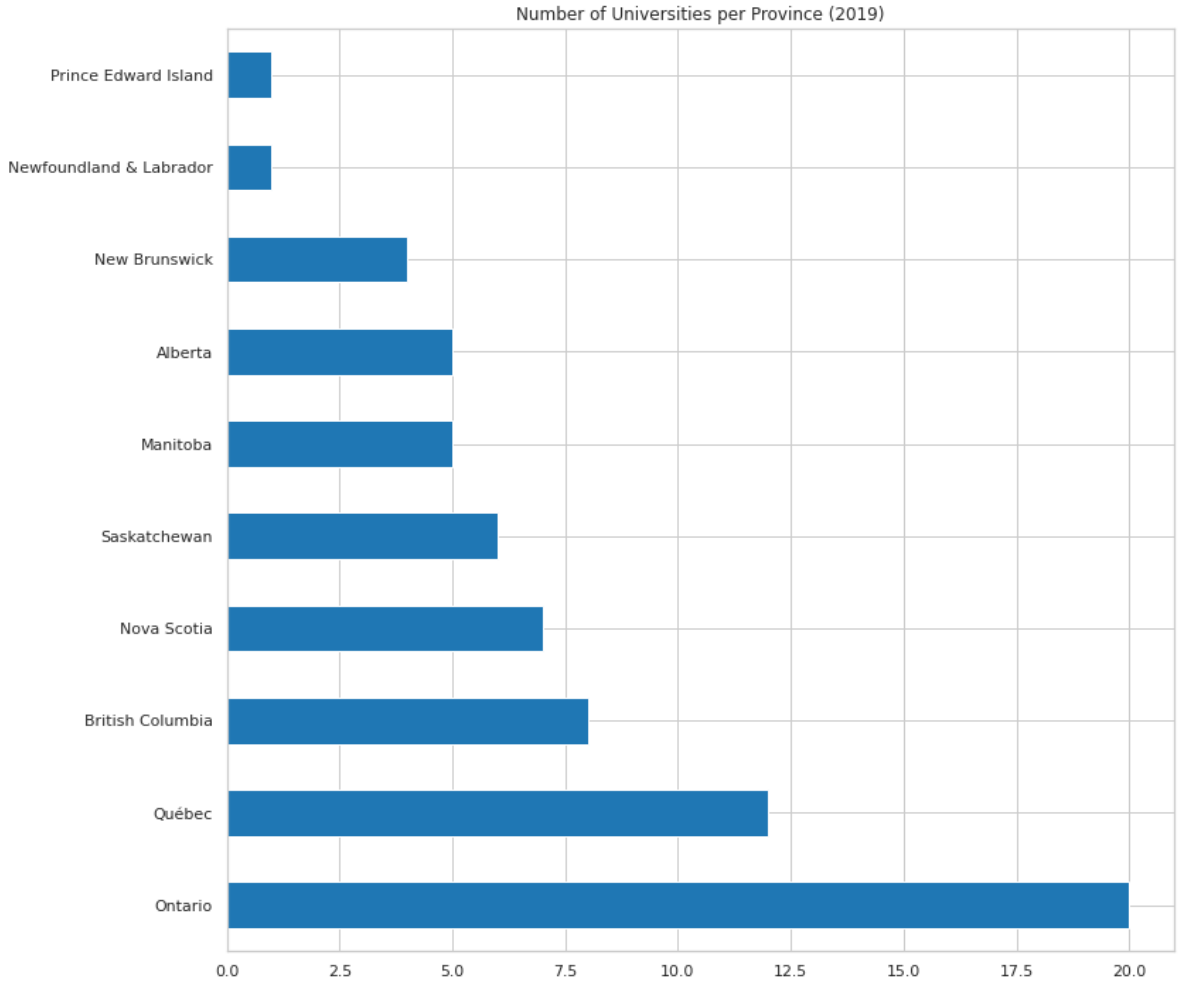


In [175]:

```
# Plotting Number of Universities per Province in 2019
df2['Province'].value_counts().plot(kind='barh', figsize=(12,12)
, legend=None)
plt.title('Number of Universities per Province (2019)')
```

Out[175]:

```
Text(0.5, 1.0, 'Number of Universities per Province
(2019)')
```



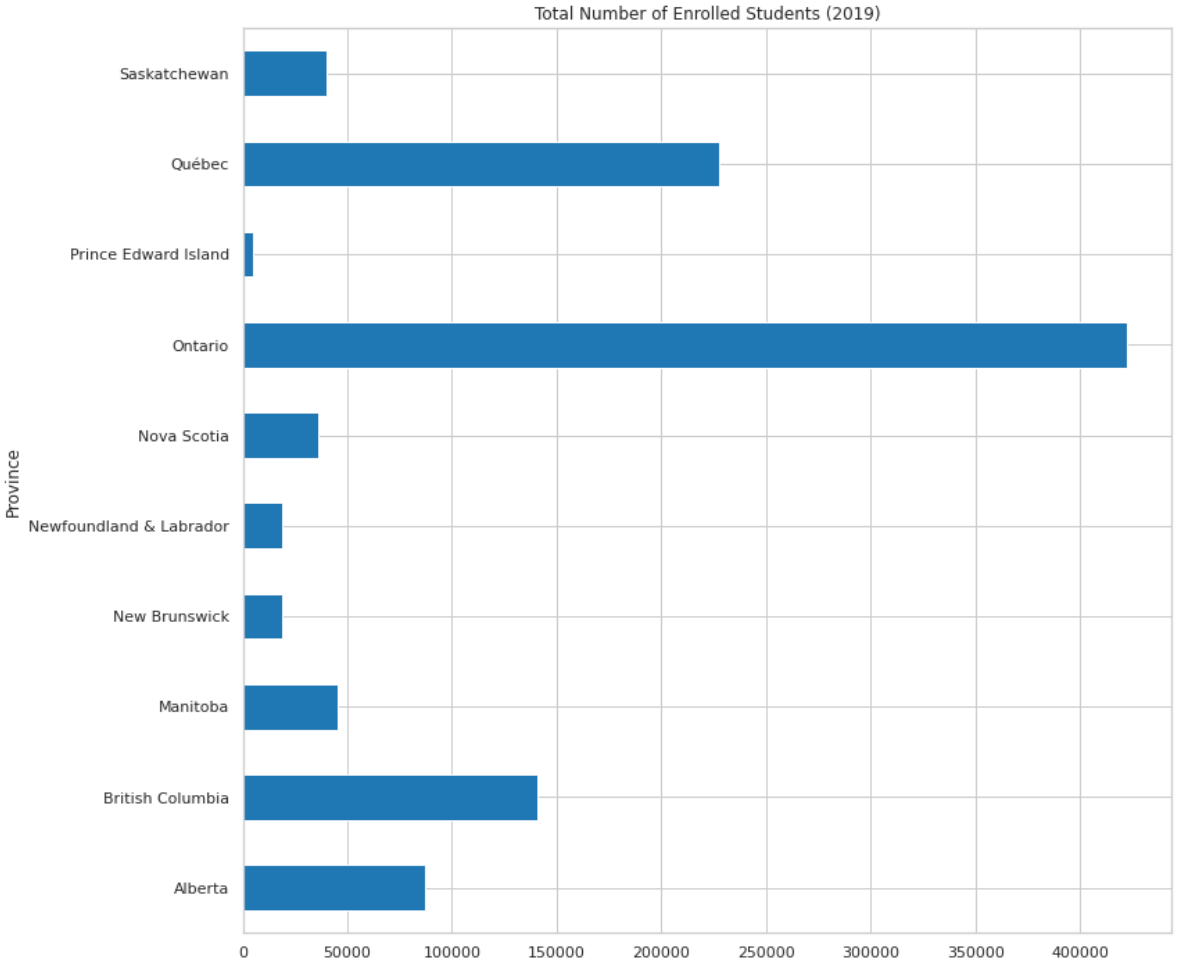
Around 48% of Canadian universities are located in Ontario and Québec.

In [174]:

```
# Plotting Total Number of Enrolled Students in 2019
df2.groupby('Province').agg({'Total': 'sum'}).plot(kind='barh',
figsize=(12,12), legend=None)
plt.title('Total Number of Enrolled Students (2019)')
```

Out[174]:

```
Text(0.5, 1.0, 'Total Number of Enrolled Students (2
019)')
```



In [141]:

```
# Total Number of Enrolled Students in 2019 (Table View)
df2.groupby('Province').agg({'Total': 'sum'})
```

Out[141]:

Province	Total
Alberta	86751
British Columbia	140895
Manitoba	45187
New Brunswick	18731
Newfoundland & Labrador	18516
Nova Scotia	35805
Ontario	422549
Prince Edward Island	4926
Québec	227299
Saskatchewan	40005

77% of all students are located in four Canadian provinces: Ontario (422,549), Québec(227,951), British Columbia(140,895) and Alberta(86,751).

In [0]:

```
# Getting access to Google Drive
from google.colab import auth
auth.authenticate_user()
import gspread
from oauth2client.client import GoogleCredentials
gc = gspread.authorize(GoogleCredentials.get_application_default
())
```

In [0]:

```
#Loaing Data
wb = gc.open_by_url('https://docs.google.com/spreadsheets/d/1MDd
vQSwrZDAnSYwXG_9yg_wBu2xHr8u_YTZVSrwANWc')
```

In [0]:

```
# Loading Alberta Tab to Data Frame
sheet_alberta = wb.worksheet('Alberta')
data_alberta = sheet_alberta.get_all_values()
df_alberta = pd.DataFrame(data_alberta)
df_alberta.columns = df_alberta.iloc[0]
df_alberta = df_alberta.iloc[1:]
```

In [58]:

```
df_alberta.tail()
```

Out[58]:

	PROVINCE	UNIVERSITY	CITY	ADDRESS	BEDS	PRICE
1721	AB	Grant MacEwan University	Edmonton	10325 115 St NW, Edmonton (North West)Alberta,...	1	795
1722	AB	Grant MacEwan University	Edmonton	10325 115 St NW, Edmonton (North West)Alberta,...	1	849
1723	AB	University of Alberta	Edmonton	9916 85 ave, Edmonton (South East)Alberta, Canada	1	890
1724	AB	University of Alberta	Edmonton	10008 86 Ave., Edmonton (South East)Alberta, C...	2	1895
1725	AB	University of Alberta	Edmonton	9911 85 Avenue, Edmonton (South East)Alberta, ...	1	975

In [0]:

```
# Cleaning and Converting PRICE, BEDS columns into integers
import re
def clean(s):
    s = s.replace('[^\d.]', '')
    return str(s)
```

In [0]:

```
# Appling clean function
for i in range(1, len(df_alberta)):
      df_alberta["PRICE"][i] = clean(df_alberta["PRICE"][i])
```

In [0]:

```
df_alberta['PRICE'] = pd.to_numeric(df_alberta['PRICE'])
df_alberta['BEDS'] = pd.to_numeric(df_alberta['BEDS'])
```

In [0]:

```
# Aggregating Data Frame by Province and University
df_alberta_agg = df_alberta.groupby(['PROVINCE','UNIVERSITY'],as
_index=False).agg({'PRICE': 'sum', 'BEDS': 'sum'}).eval('AVG_PRI
CE = PRICE / BEDS')
```

In [0]:

```
# Getting Median Price
m = df_alberta.groupby('UNIVERSITY')['PRICE'].median()
df_alberta_agg['MEDIAN_PRICE'] = m.values
```

In [107]:

```
# Getting Table View
df_alberta_agg.head()
```

Out[107]:

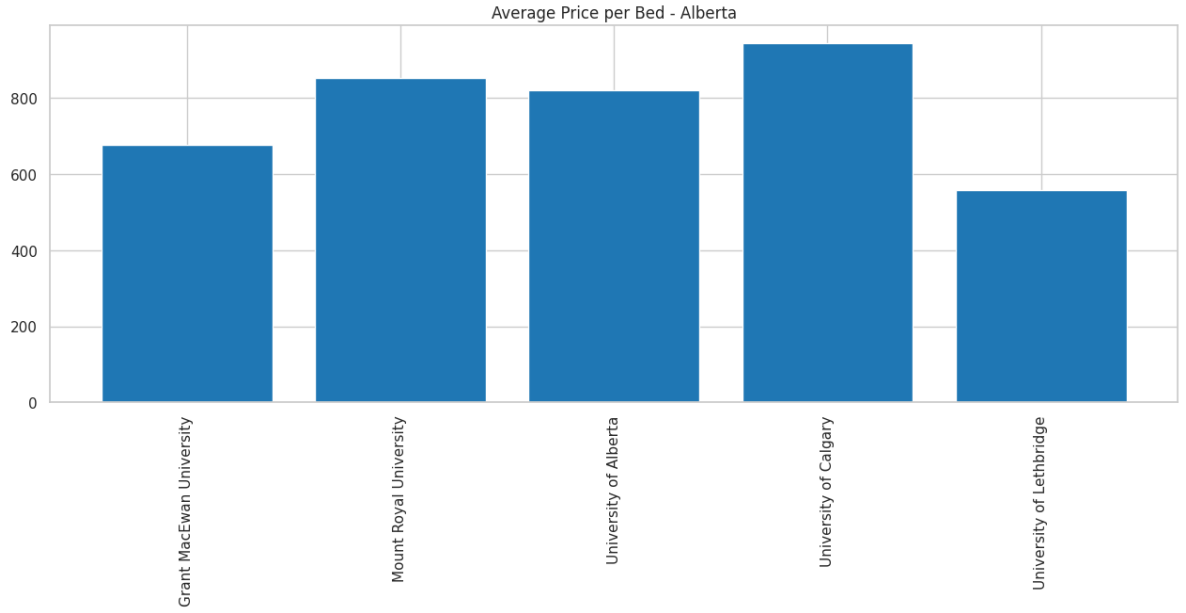
	PROVINCE	UNIVERSITY	PRICE	BEDS	AVG_PRICE	MEDIAN_PRIC
0	AB	Grant MacEwan University	1164117	1723	675.633778	975.
1	AB	Mount Royal University	43390	51	850.784314	1125.
2	AB	University of Alberta	213577	261	818.302682	1025.
3	AB	University of Calgary	536869	570	941.875439	1250.
4	AB	University of Lethbridge	6689	12	557.416667	962.

In [0]:

```
# Get the number of international students for each university i
n the beds data set and append the column
# df_alberta_combined = df_alberta_agg.merge(univ_by_prov, on='U
niversity')
```

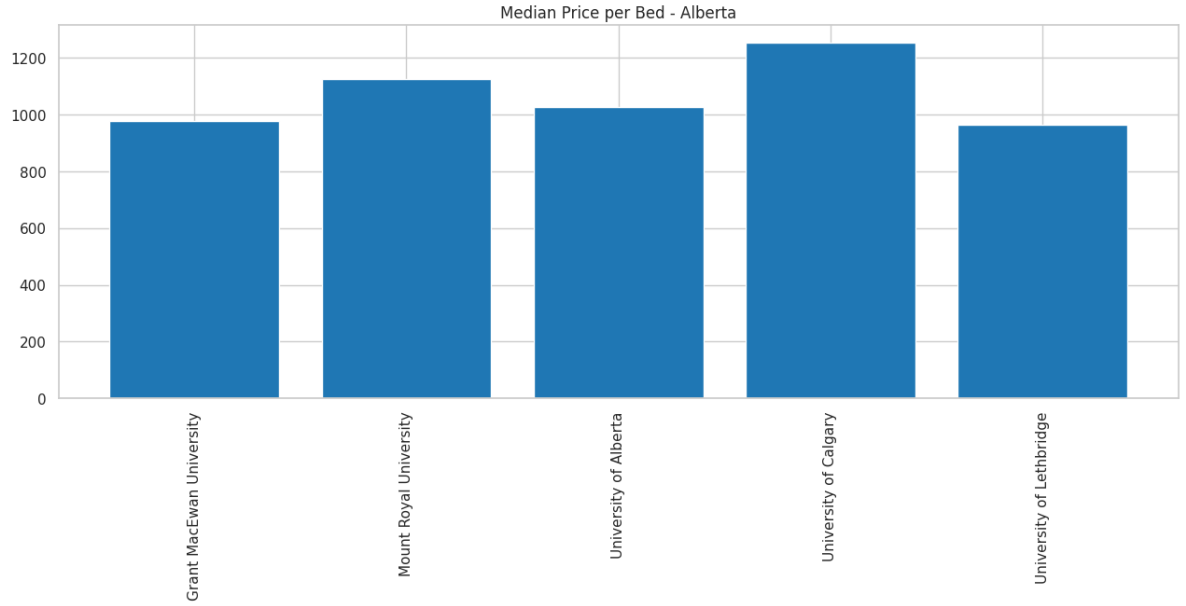
In [0]:

```
# Plotting Average Price per Bed in Alberta
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure
fig = plt.figure(dpi=100)
ax = fig.add_axes([0,0,2,1])
ax.bar(df_alberta_agg.UNIVERSITY,df_alberta_agg.AVG_PRICE)
plt.xticks(rotation=90)
plt.title('Average Price per Bed - Alberta')
plt.show()
```



In [108]:

```
# Plotting Median Price per Bed in Alberta
fig = plt.figure(dpi=100)
ax = fig.add_axes([0,0,2,1])
ax.bar(df_alberta_agg.UNIVERSITY,df_alberta_agg.MEDIAN_PRICE)
plt.xticks(rotation=90)
plt.title('Median Price per Bed - Alberta')
plt.show()
```



In [0]:

```
df_alberta_beds = df_alberta.groupby(['PROVINCE','UNIVERSITY'],as_index=False).agg({'BEDS': 'count'})
```

In [0]:

```
df_alberta_beds = df_alberta_beds.rename(columns={'UNIVERSITY': 'University'})
```

In [0]:

```
df_alberta_beds.head()
```

Out[0]:

	PROVINCE	University	BEDS
0	AB	Grant MacEwan University	1105
1	AB	Mount Royal University	38
2	AB	University of Alberta	201
3	AB	University of Calgary	373
4	AB	University of Lethbridge	8

In [0]:

```
AB_int_dom.head()
```

Out[0]:

	Province	University	Total	International	Domestic
30	Alberta	University of Lethbridge	8956	1044.069254	7911.930746
51	Alberta	Concordia University of Edmonton	2771	323.036613	2447.963387
65	Alberta	The King's University	850	99.090986	750.909015
5	Alberta	University of Alberta	39938	4655.877387	35282.122613
10	Alberta	University of Calgary	34236	3991.151741	30244.848259

In [0]:

```
# Mearging AB_int_dom and df_alberta_beds data frames
new_table_ab = pd.merge(df_alberta_beds, AB_int_dom, on='Univers
ity', how='inner')
```

In [0]:

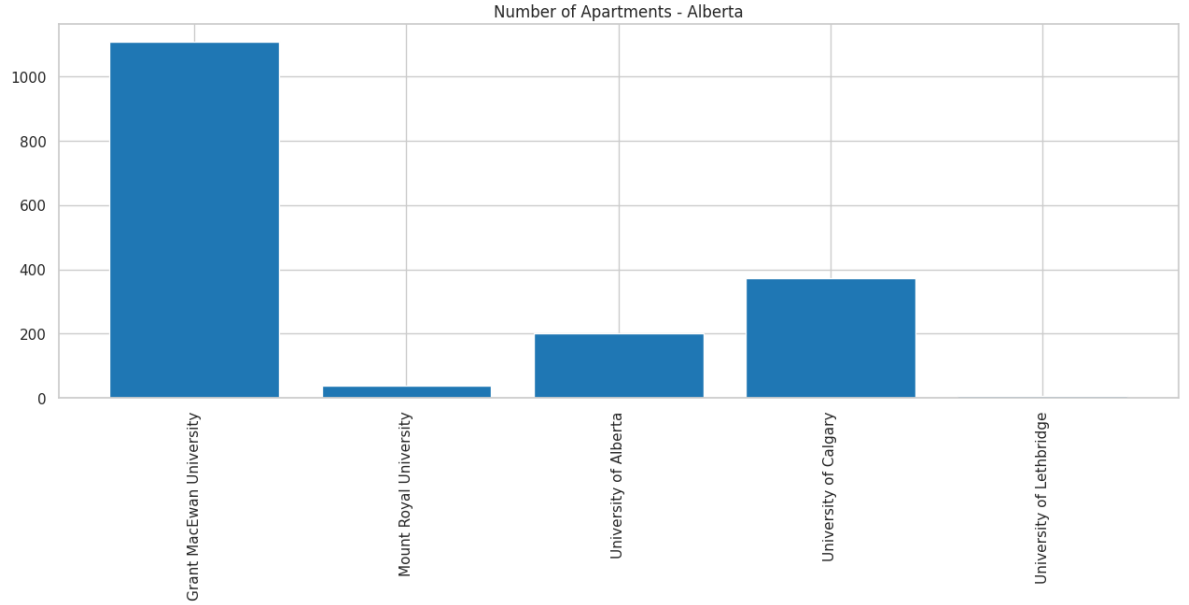
```
new_table_ab.head()
#new_table_ab.International = new_table_ab.International.astype(
float)
```

Out[0]:

	PROVINCE	University	BEDS	Province	Total	International	Don
0	AB	University of Alberta	201	Alberta	39938	4655.877387	35282.1
1	AB	University of Calgary	373	Alberta	34236	3991.151741	30244.8
2	AB	University of Lethbridge	8	Alberta	8956	1044.069254	7911.9

In [0]:

```
# Plotting Number of Available Apartments in Alberta
fig = plt.figure(dpi=100)
ax = fig.add_axes([0,0,2,1])
ax.bar(df_alberta_beds.University,df_alberta_beds.BEDS)
plt.xticks(rotation=90)
plt.title('Number of Apartments - Alberta')
plt.show()
```

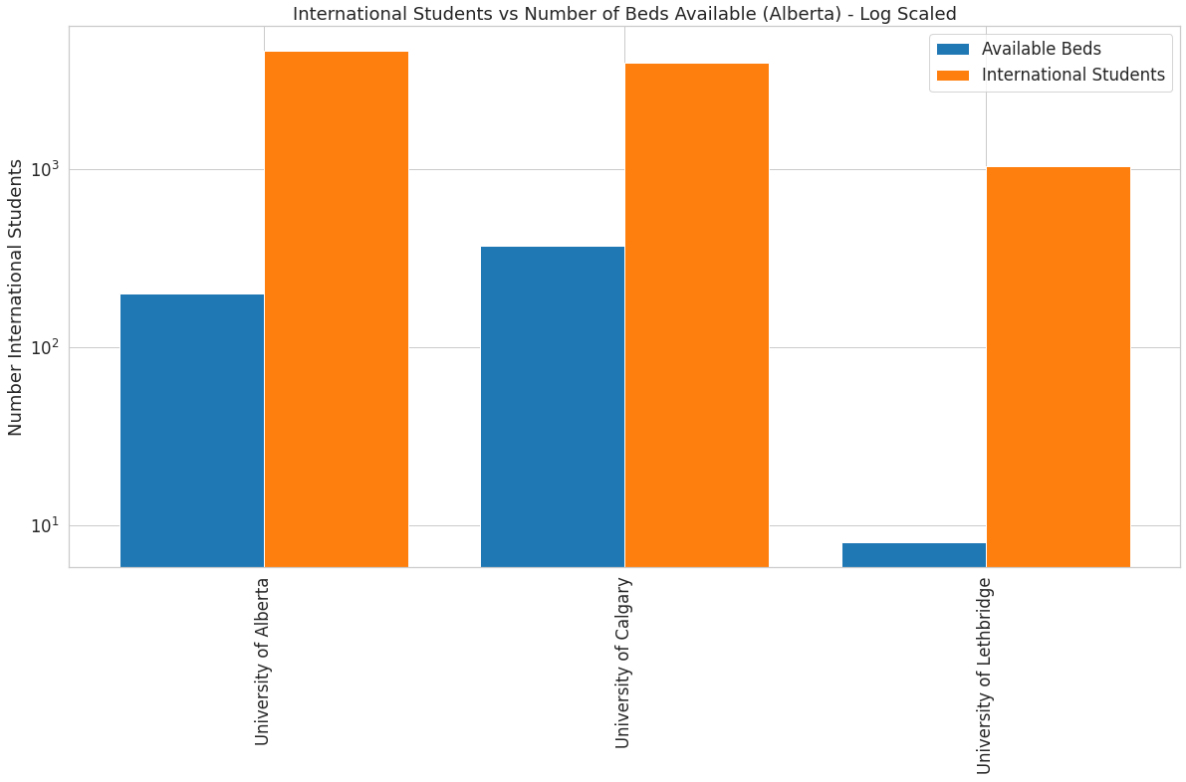


In [0]:

```
# Plotting Alberta International Students vs Number of Beds Avai
lable
x = np.arange(len(new_table_ab.University))  # the label locatio
ns
width = 0.4  # the width of the bars
labels = new_table_ab.University
fig, ax = plt.subplots()
rects1 = ax.bar(x - width/2, new_table_ab.BEDS, width, label='Av
ailable Beds')
rects2 = ax.bar(x + width/2, new_table_ab.International, width,
label='International Students')
ax.set_yscale('log')
fig.set_size_inches(20,10)
ax.set_ylabel('Number International Students')
ax.set_title('International Students vs Number of Beds Available
(Alberta) - Log Scaled')
ax.set_xticks(x)
ax.set_xticklabels(labels)
plt.xticks(rotation=90)
ax.legend()
```

Out[0]:

```
<matplotlib.legend.Legend at 0x7fda6cdbccc0>
```



In [0]:

```
# Graphs above indicate a possible demand in off-campus student housing in Alberta.
# For example, University of Alberta has the highest price per bed and only 154 available apartments and around 4700 international students.
# Appartments around University of Alberta have the highest price per bed ($884) that can indicate higher demand.
```

In [0]:

```
# Loading British Columbia Tab to Data Frame
sheet_bc = wb.worksheet('British Columbia')
data_bc = sheet_bc.get_all_values()
df_bc = pd.DataFrame(data_bc)
df_bc.columns = df_bc.iloc[0]
df_bc = df_bc.iloc[1:]
```

In [111]:

```
df_bc.tail()
```

Out[111]:

	PROVINCE	UNIVERSITY	CITY	ADDRESS	BEDS	PRICE	SOU
189	BC	Thompson Rivers University	Kamloops	960 13th St	1	550	ki
190	BC	Kwantlen Polytechnic University	Surrey	15945 96 Ave	1	650	ki
191	BC	University of British Columbia - Okanagan	Kelowna	2234 Quail Run Dr	1	750	ki
192	BC	University of British Columbia	Vancouver	1311 Howe St	1	1100	ki
193	BC	University of Victoria	Victoria	Blair Ave	1	768	ki

In [0]:

```
# Appling clean function
for i in range(1, len(df_bc)):
      df_bc["PRICE"][i] = clean(df_bc["PRICE"][i])
```

In [0]:

```
df_bc['PRICE'] = pd.to_numeric(df_bc['PRICE'])
df_bc['BEDS'] = pd.to_numeric(df_bc['BEDS'])
```

In [0]:

```
# Aggregating Data Frame by Province and University
df_bc_agg = df_bc.groupby(['PROVINCE','UNIVERSITY'],as_index=False).agg({'PRICE': 'sum', 'BEDS': 'sum'}).eval('AVG_PRICE = PRICE / BEDS')
```

In [0]:

```
# Getting Median Price
m = df_bc.groupby('UNIVERSITY')['PRICE'].median()
df_bc_agg['MEDIAN_PRICE'] = m.values
```

In [116]:

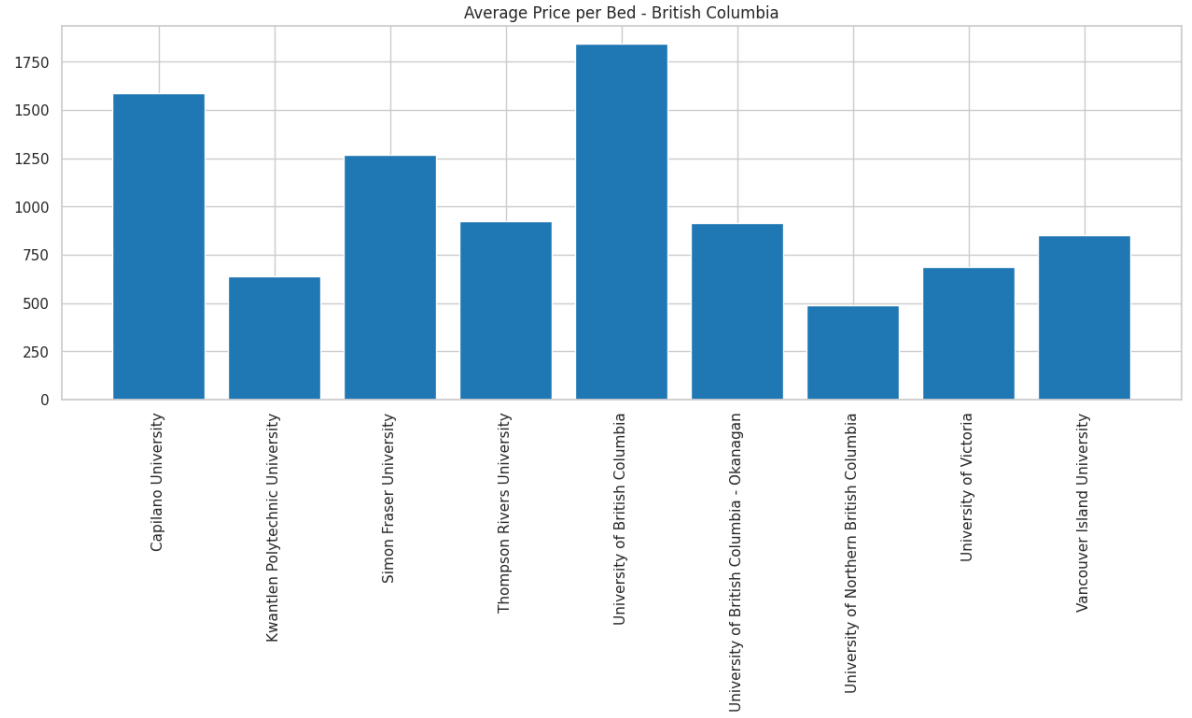
```
df_bc_agg
```

Out[116]:

	PROVINCE	UNIVERSITY	PRICE	BEDS	AVG_PRICE	MEDIAN_PRICE
0	BC	Capilano University	103806	65.5	1584.824427	2500.0
1	BC	Kwantlen Polytechnic University	1275	2.0	637.500000	637.5
2	BC	Simon Fraser University	110154	87.0	1266.137931	1750.5
3	BC	Thompson Rivers University	28628	31.0	923.483871	1342.0
4	BC	University of British Columbia	37775	20.5	1842.682927	1800.0
5	BC	University of British Columbia - Okanagan	30245	33.0	916.515152	1377.5
6	BC	University of Northern British Columbia	980	2.0	490.000000	490.0
7	BC	University of Victoria	10321	15.0	688.066667	700.0
8	BC	Vancouver Island University	12785	15.0	852.333333	1250.0

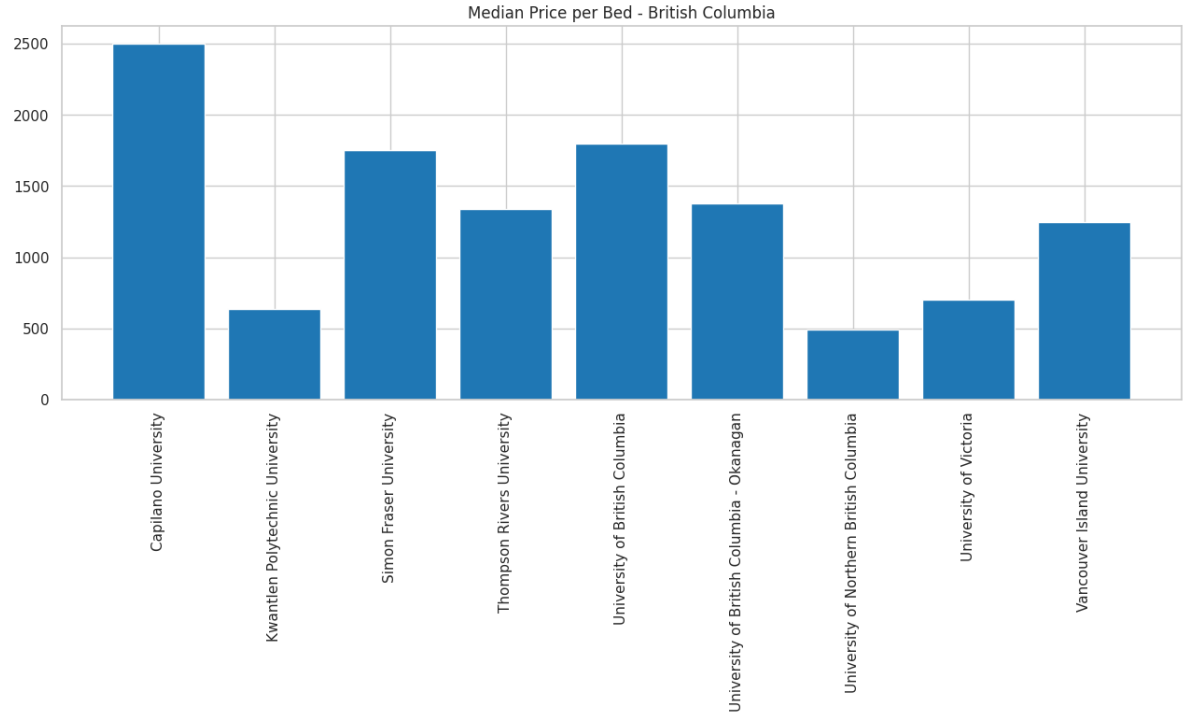
In [0]:

```
# Plotting Average Price per Bed in British Columbia
import matplotlib.pyplot as plt
fig = plt.figure(dpi=100)
ax = fig.add_axes([0,0,2,1])
ax.bar(df_bc_agg.UNIVERSITY,df_bc_agg.AVG_PRICE)
plt.xticks(rotation=90)
plt.title('Average Price per Bed - British Columbia')
plt.show()
```



In [117]:

```
# Plotting Median Price per Bed in British Columbia
fig = plt.figure(dpi=100)
ax = fig.add_axes([0,0,2,1])
ax.bar(df_bc_agg.UNIVERSITY,df_bc_agg.MEDIAN_PRICE)
plt.xticks(rotation=90)
plt.title('Median Price per Bed - British Columbia')
plt.show()
```



In [0]:

```
df_bc_beds = df_bc.groupby(['PROVINCE','UNIVERSITY'],as_index=False).agg({'BEDS': 'count'})
```

In [0]:

```
df_bc_beds.head()
```

Out[0]:

	PROVINCE	UNIVERSITY	BEDS
0	BC	Capilano University	40
1	BC	Kwantlen Polytechnic University	2
2	BC	Simon Fraser University	60
3	BC	Thompson Rivers University	22
4	BC	University of British Columbia	17

In [0]:

```
BC_int_dom
```

Out[0]:

	Province	University	Total	International	Domestic
14	British Columbia	Simon Fraser University	30254	7363.025802	22890.974198
48	British Columbia	University of Northern British Columbia	3444	838.178782	2605.821218
28	British Columbia	Thompson Rivers University	9595	2335.169980	7259.830020
55	British Columbia	Emily Carr University of Art + Design	1893	460.706282	1432.293718
44	British Columbia	Trinity Western University	4434	1079.118675	3354.881325
33	British Columbia	Vancouver Island University	7594	1848.179346	5745.820654
1	British Columbia	The University of British Columbia	61547	14978.916806	46568.083194
17	British Columbia	University of Victoria	22134	5386.831926	16747.168074

In [0]:

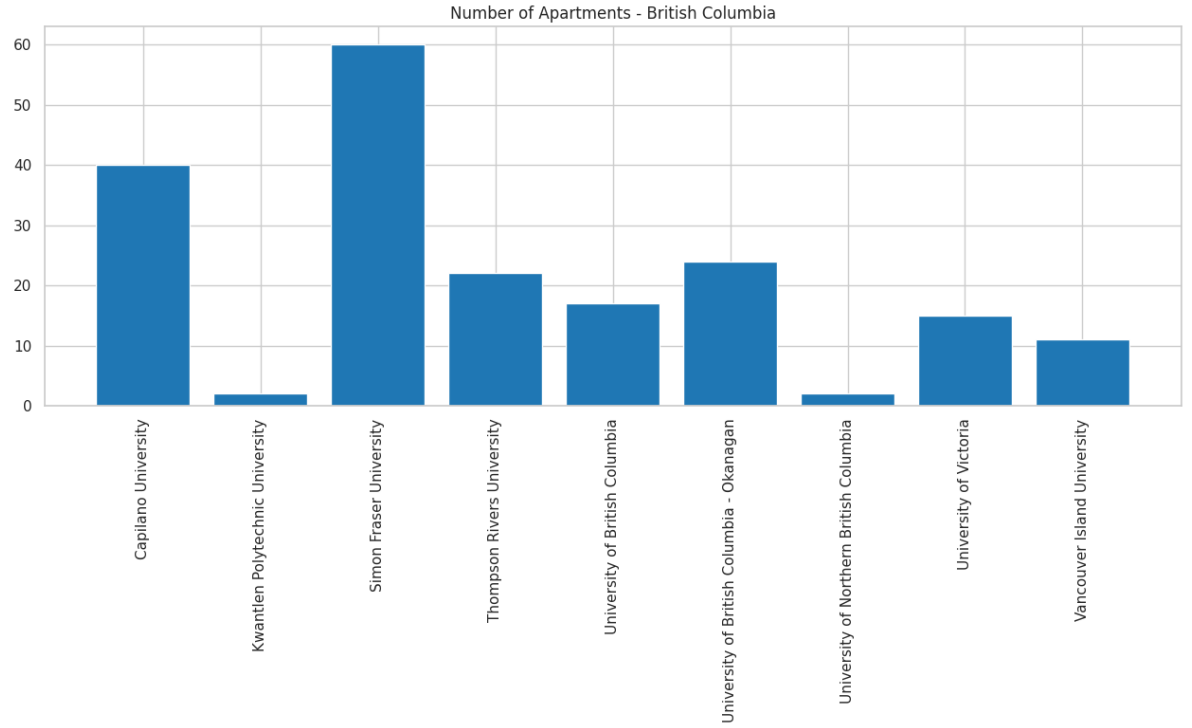
```
# Mearging BC_int_dom and df_bc_beds data frames
df_bc_beds = df_bc_beds.rename(columns={'UNIVERSITY': 'University'})
new_table_bc = pd.merge(df_bc_beds, BC_int_dom, on='University', how='inner')
new_table_bc.head(n=10)
```

Out[0]:

	PROVINCE	University	BEDS	Province	Total	International	Don
0	BC	Simon Fraser University	60	British Columbia	30254	7363.025802	22890.9
1	BC	Thompson Rivers University	22	British Columbia	9595	2335.169980	7259.8
2	BC	University of Northern British Columbia	2	British Columbia	3444	838.178782	2605.8
3	BC	University of Victoria	15	British Columbia	22134	5386.831926	16747.1
4	BC	Vancouver Island University	11	British Columbia	7594	1848.179346	5745.8

In [0]:

```
# Plotting Number of Available Apartments in British Columbia
import matplotlib.pyplot as plt
fig = plt.figure(dpi=100)
ax = fig.add_axes([0,0,2,1])
ax.bar(df_bc_beds.University,df_bc_beds.BEDS)
plt.xticks(rotation=90)
plt.title('Number of Apartments - British Columbia')
plt.show()
```

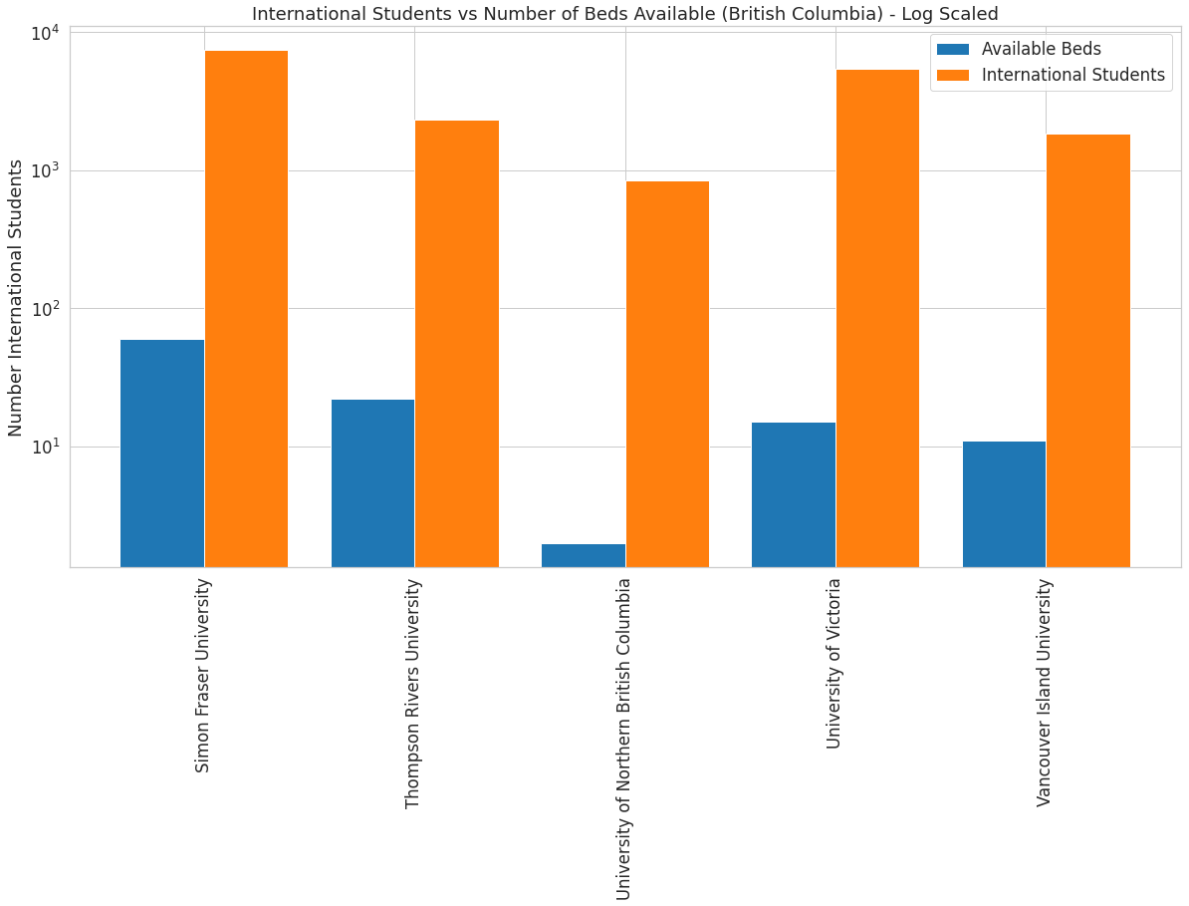


In [0]:

```
# Plotting British Columbia International Students vs Number of Beds Available

x = np.arange(len(new_table_bc.University))  # the label locations
width = 0.4  # the width of the bars
labels = new_table_bc.University
fig, ax = plt.subplots()
rects1 = ax.bar(x - width/2, new_table_bc.BEDS, width, label='Available Beds')
rects2 = ax.bar(x + width/2, new_table_bc.International, width, label='International Students')
ax.set_yscale('log')
fig.set_size_inches(20,10)
ax.set_ylabel('Number International Students')
ax.set_title('International Students vs Number of Beds Available (British Columbia) - Log Scaled')
ax.set_xticks(x)
ax.set_xticklabels(labels)
plt.xticks(rotation=90)
ax.legend()
```

Out[0]:

```
<matplotlib.legend.Legend at 0x7fda6d0a6da0>
```



In [0]:

```
# BC might lack off-campus student housing since the province has a high percentage of international students and limited apartment availability.
# All Universities have a high price per bed.
```

In [0]:

```
# Loading Ontario Tab to Data Frame
sheet_on = wb.worksheet('Ontario')
data_on = sheet_on.get_all_values()
df_on = pd.DataFrame(data_on)
df_on.columns = df_on.iloc[0]
df_on = df_on.iloc[1:]
```

In [120]:

```
df_on.tail()
```

Out[120]:

	PROVINCE	UNIVERSITY	CITY	ADDRESS	BEDS	PRICE	SOUI
2330	ON	York University	Toronto	Cook Rd, North York, ON M3J 3T2	1	850	kijii
2331	ON	York University	Toronto	Cook Rd, North York, ON M3J 3T2	1	850	kijii
2332	ON	York University	Toronto	4 Aldwinckle Heights, North York, ON M3J 3S6	1	850	kijii
2333	ON	York University	Toronto	143 Holmes Ave, North York, ON M2N 4M5	1	920	kijii
2334	ON	York University	Toronto	Toronto M3h2y2 ON	1	950	kijii

In [0]:

```
# Appling clean function
for i in range(1, len(df_on)):
      df_on["PRICE"][i] = clean(df_on["PRICE"][i])
```

In [0]:

```
df_on['PRICE'] = pd.to_numeric(df_on['PRICE'])
df_on['BEDS'] = pd.to_numeric(df_on['BEDS'])
```

In [0]:

```
# Aggregating Data Frame by Province and University
df_on_agg = df_on.groupby(['PROVINCE','UNIVERSITY'],as_index=False).agg({'PRICE': 'sum', 'BEDS': 'sum'}).eval('AVG_PRICE = PRICE / BEDS')
```

In [0]:

```
# Getting Median Price
m = df_on.groupby('UNIVERSITY')['PRICE'].median()
df_on_agg['MEDIAN_PRICE'] = m.values
```

In [125]:

```
df_on_agg
```

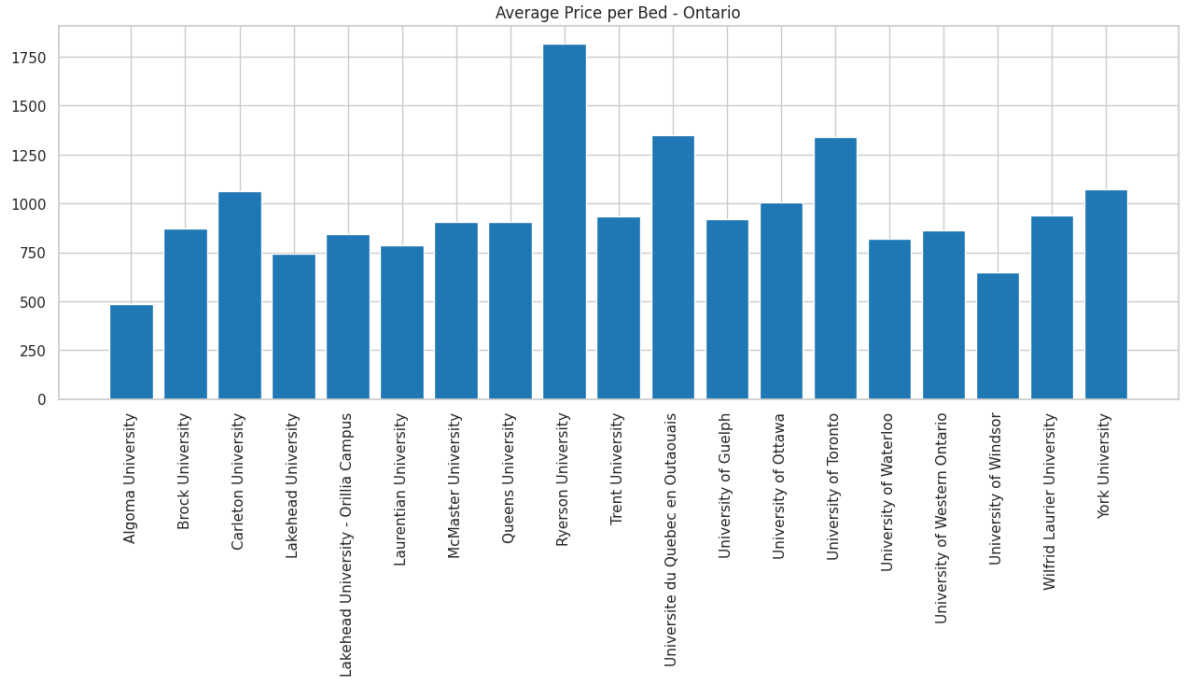
Out[125]:

	PROVINCE	UNIVERSITY	PRICE	BEDS	AVG_PRICE	MEDIAN_P
0	ON	Algoma University	2425.00	5.0	485.000000	
1	ON	Brock University	119667.00	137.0	873.481752	
2	ON	Carleton University	474999.00	447.0	1062.637584	
3	ON	Lakehead University	16302.00	22.0	741.000000	
4	ON	Lakehead University - Orillia Campus	16818.00	20.0	840.900000	
5	ON	Laurentian University	58820.00	75.0	784.266667	
6	ON	McMaster University	222720.00	246.0	905.365854	
7	ON	Queens University	81682.00	90.0	907.577778	
8	ON	Ryerson University	1233452.67	678.0	1819.251726	2

9	ON	Trent University	9330.00	10.0	933.000000	-
10	ON	Universite du Quebec en Outaouais	116209.00	86.0	1351.267442	-
11	ON	University of Guelph	93586.00	102.0	917.509804	-
12	ON	University of Ottawa	123725.00	123.0	1005.894309	-
13	ON	University of Toronto	367951.00	275.0	1338.003636	-
14	ON	University of Waterloo	199154.00	242.5	821.253608	-
15	ON	University of Western Ontario	403807.00	468.0	862.835470	-
16	ON	University of Windsor	26615.00	41.0	649.146341	
17	ON	Wilfrid Laurier University	144829.00	154.0	940.448052	-
18	ON	York University	211929.00	198.0	1070.348485	-

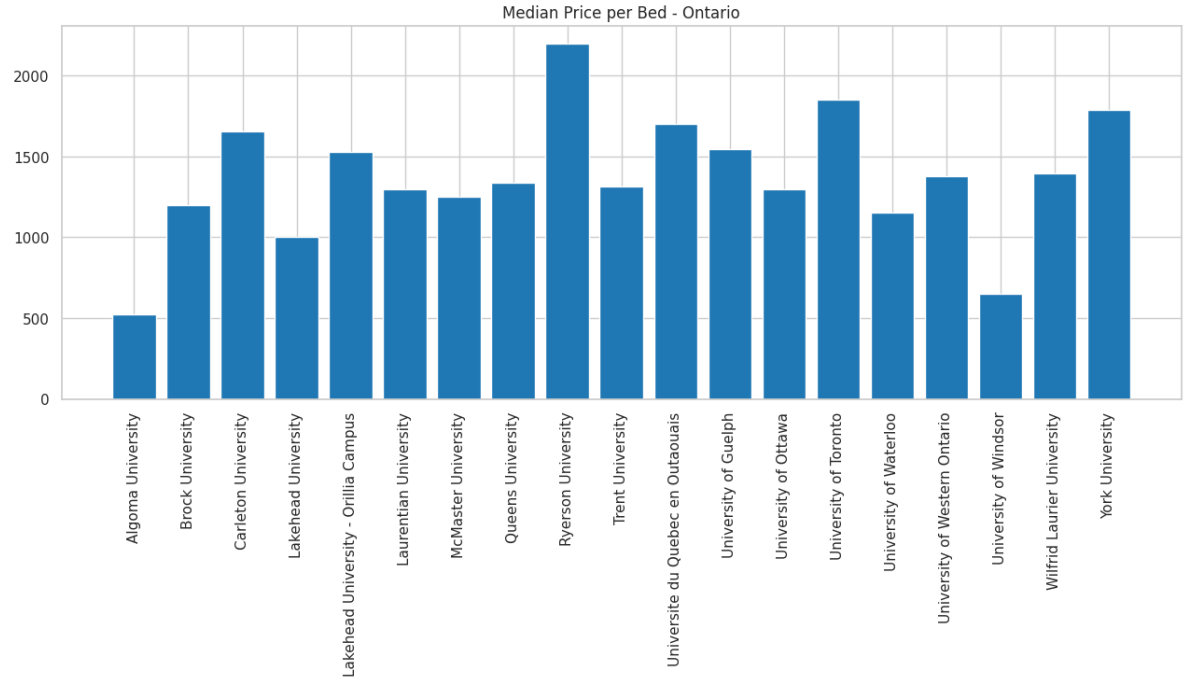
In [0]:

```
# Plotting Average Price per Bed in Ontario
import matplotlib.pyplot as plt
fig = plt.figure(dpi=100)
ax = fig.add_axes([0,0,2,1])
ax.bar(df_on_agg.UNIVERSITY,df_on_agg.AVG_PRICE)
plt.xticks(rotation=90)
plt.title('Average Price per Bed - Ontario')
plt.show()
```



In [126]:

```
# Plotting Median Average Price per Bed in Ontario
fig = plt.figure(dpi=100)
ax = fig.add_axes([0,0,2,1])
ax.bar(df_on_agg.UNIVERSITY,df_on_agg.MEDIAN_PRICE)
plt.xticks(rotation=90)
plt.title('Median Price per Bed - Ontario')
plt.show()
```



In [0]:

```
df_on_beds = df_on.groupby(['PROVINCE','UNIVERSITY'],as_index=False).agg({'BEDS': 'count'})
```

In [0]:

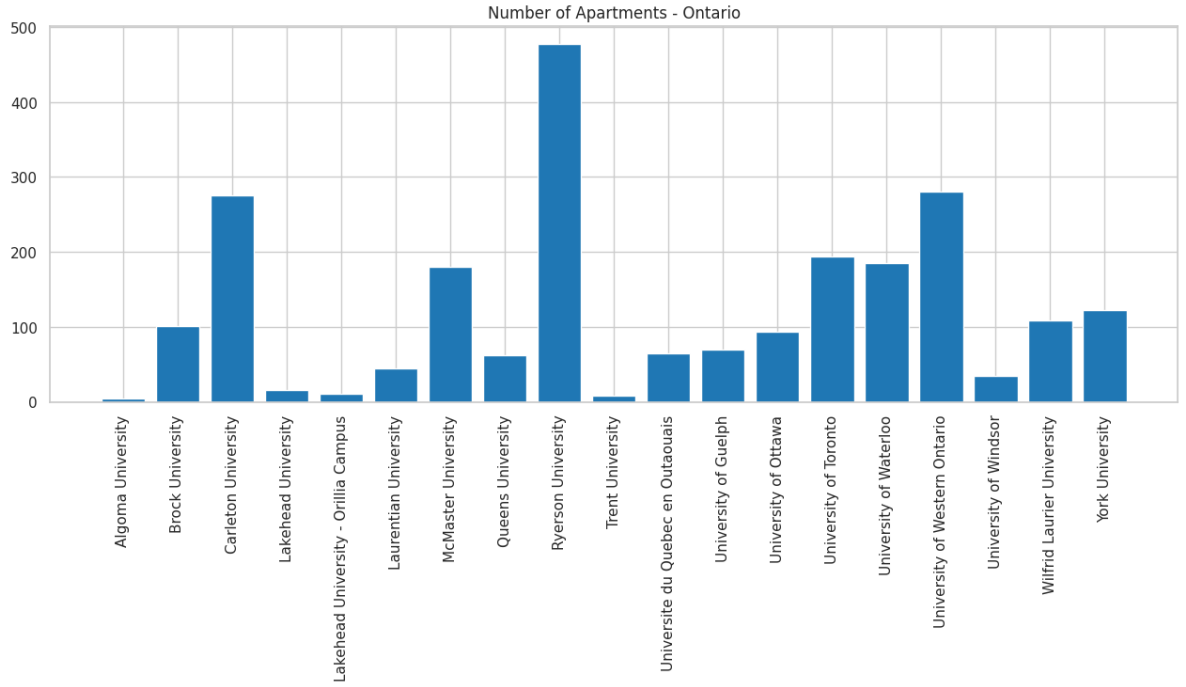
```
df_on_beds.head(n=20)
```

Out[0]:

	PROVINCE	UNIVERSITY	BEDS
0	ON	Algoma University	5
1	ON	Brock University	101
2	ON	Carleton University	275
3	ON	Lakehead University	16
4	ON	Lakehead University - Orillia Campus	11
5	ON	Laurentian University	44
6	ON	McMaster University	180
7	ON	Queens University	62
8	ON	Ryerson University	478
9	ON	Trent University	8
10	ON	Universite du Quebec en Outaouais	65
11	ON	University of Guelph	70
12	ON	University of Ottawa	93
13	ON	University of Toronto	194
14	ON	University of Waterloo	185
15	ON	University of Western Ontario	281
16	ON	University of Windsor	35
17	ON	Wilfrid Laurier University	108
18	ON	York University	123

In [0]:

```
# Plotting Number of Available Apartments in Ontario
import matplotlib.pyplot as plt
fig = plt.figure(dpi=100)
ax = fig.add_axes([0,0,2,1])
ax.bar(df_on_beds.UNIVERSITY,df_on_beds.BEDS)
plt.xticks(rotation=90)
plt.title('Number of Apartments - Ontario')
plt.show()
```



In [0]:

```
# Mearging ON_int_dom and df_on_beds data frames

df_on_beds = df_on_beds.rename(columns={'UNIVERSITY': 'University'})
new_table_on = pd.merge(df_on_beds, ON_int_dom, on='University', how='inner')
new_table_on.head(n=10)
```

Out[0]:

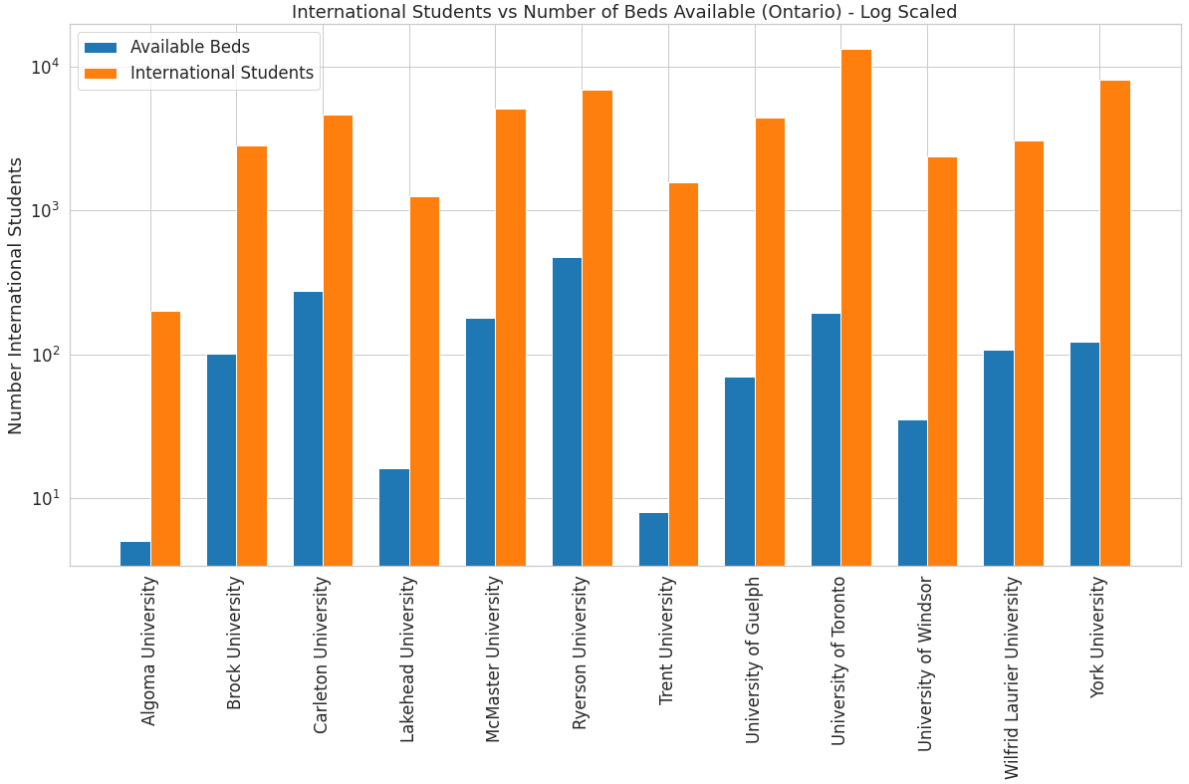
	PROVINCE	University	BEDS	Province	Total	International	Dor
0	ON	Algoma University	5	Ontario	1370	199.774386	1170.2
1	ON	Brock University	101	Ontario	19560	2852.253283	16707.7
2	ON	Carleton University	275	Ontario	31790	4635.640689	27154.3
3	ON	Lakehead University	16	Ontario	8620	1256.974606	7363.0
4	ON	McMaster University	180	Ontario	35040	5109.558029	29930.4
5	ON	Ryerson University	478	Ontario	47350	6904.611092	40445.3
6	ON	Trent University	8	Ontario	10880	1586.529434	9293.4
7	ON	University of Guelph	70	Ontario	30310	4419.826023	25890.1
8	ON	University of Toronto	194	Ontario	92000	13415.506240	78584.4
9	ON	University of Windsor	35	Ontario	16480	2403.125466	14076.8

In [0]:

```
# Plotting Ontario International Students vs Number of Beds Avai
lable

x = np.arange(len(new_table_on.University))  # the label locatio
ns
width = 0.35  # the width of the bars
labels = new_table_on.University
fig, ax = plt.subplots()
rects1 = ax.bar(x - width/2, new_table_on.BEDS, width, label='Av
ailable Beds')
rects2 = ax.bar(x + width/2, new_table_on.International, width,
label='International Students')
ax.set_yscale('log')
fig.set_size_inches(20,10)
ax.set_ylabel('Number International Students')
ax.set_title('International Students vs Number of Beds Available
(Ontario) - Log Scaled')
ax.set_xticks(x)
ax.set_xticklabels(labels)
plt.xticks(rotation=90)
ax.legend()
```

Out[0]:

```
<matplotlib.legend.Legend at 0x7fda6dcc5828>
```



In [0]:

```
# Loading Quebec Tab to Data Frame
sheet_qc = wb.worksheet('Quebec')
data_qc = sheet_qc.get_all_values()
df_qc = pd.DataFrame(data_qc)
df_qc.columns = df_qc.iloc[0]
df_qc = df_qc.iloc[1:]
```

In [128]:

```
df_qc.tail()
```

Out[128]:

	PROVINCE	UNIVERSITY	CITY	ADDRESS	BEDS	PRICE	S
1661	QC	Universite Du Quebec A Montreal	Montreal	2525 Cavendish blvd.	2	1500	rentl
1662	QC	Universite Du Quebec A Montreal	Montreal	2525 Cavendish blvd.	1	820	rentl
1663	QC	Universite Du Quebec A Montreal	Montreal	2525 Cavendish blvd.	1	990	rentl
1664	QC	Universite Du Quebec A Montreal	Montreal	2500 Cavendish blvd.	1	905	rentl
1665	QC	Universite Du Quebec A Montreal	Montreal	2500 Cavendish blvd.	1	905	rentl

In [0]:

```
# Appling clean function
for i in range(1, len(df_qc)):
      df_qc["PRICE"][i] = clean(df_qc["PRICE"][i])
```

In [0]:

```
df_qc['PRICE'] = pd.to_numeric(df_qc['PRICE'])
df_qc['BEDS'] = pd.to_numeric(df_qc['BEDS'])
```

In [0]:

```
# Aggregating Data Frame by Province and University
df_qc_agg = df_qc.groupby(['PROVINCE','UNIVERSITY'],as_index=Fal
se).agg({'PRICE': 'sum', 'BEDS': 'sum'}).eval('AVG_PRICE = PRICE
/ BEDS')
```

In [0]:

```
# Getting Median Price
m = df_qc.groupby('UNIVERSITY')['PRICE'].median()
df_qc_agg['MEDIAN_PRICE'] = m.values
```

In [133]:

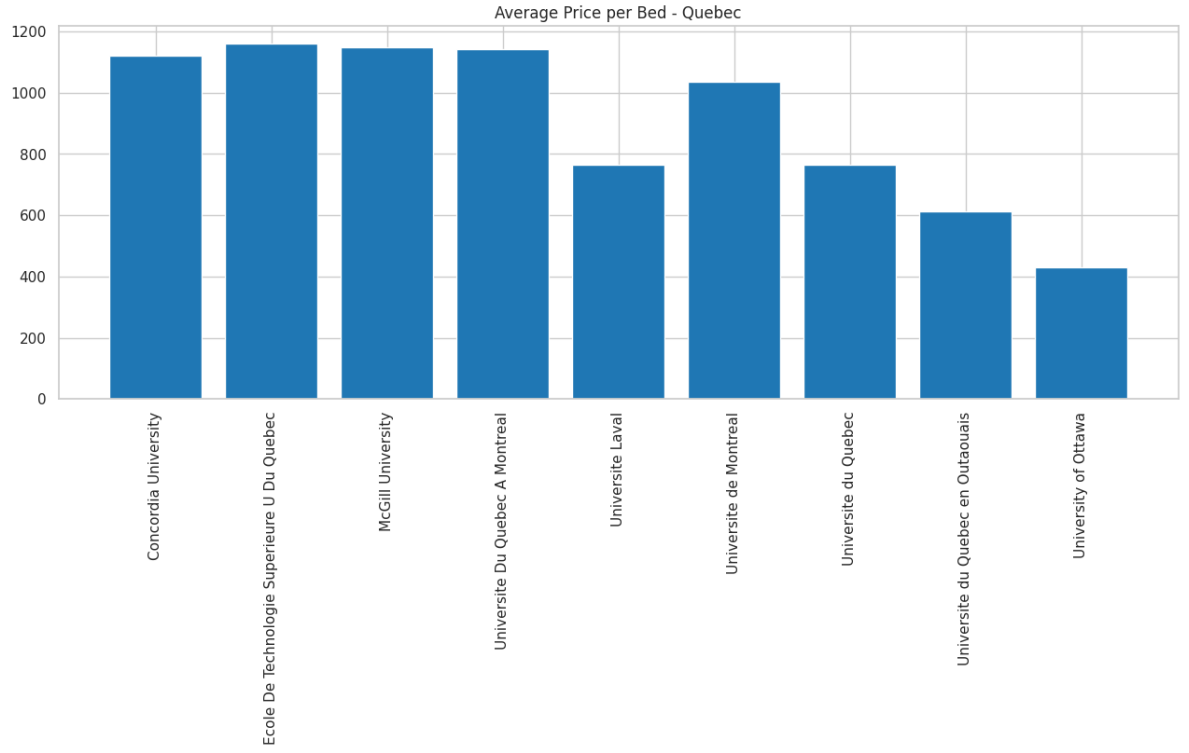
```
df_qc_agg = df_qc_agg.dropna()
df_qc_agg
```

Out[133]:

	PROVINCE	UNIVERSITY	PRICE	BEDS	AVG_PRICE	MEDIAN_PRICE
0	QC	Concordia University	623601	556	1121.584532	1477.0
1	QC	Ecole De Technologie Superieure U Du Quebec	561503	484	1160.130165	1495.0
2	QC	McGill University	581166	506	1148.549407	1495.0
3	QC	Universite Du Quebec A Montreal	606928	531	1142.990584	1499.0
4	QC	Universite Laval	57405	75	765.400000	1025.0
5	QC	Universite de Montreal	248760	240	1036.500000	1322.5
6	QC	Universite du Quebec	56645	74	765.472973	1026.0
7	QC	Universite du Quebec en Outaouais	4278	7	611.142857	1074.0
8	QC	University of Ottawa	2148	5	429.600000	1074.0

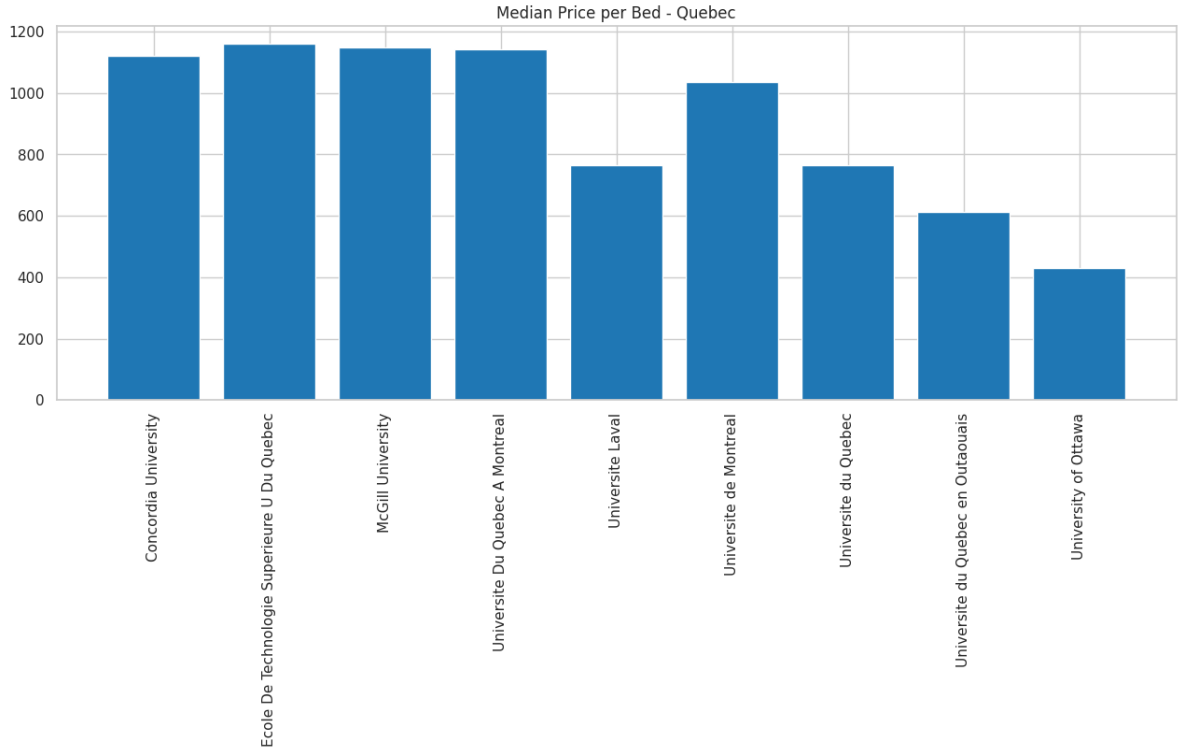
In [0]:

```
# Plotting Average Price per Bed in Quebec

import matplotlib.pyplot as plt
fig = plt.figure(dpi=100)
ax = fig.add_axes([0,0,2,1])
ax.bar(df_qc_agg.UNIVERSITY,df_qc_agg.AVG_PRICE)
plt.xticks(rotation=90)
plt.title('Average Price per Bed - Quebec')
plt.show()
```



In [134]:

```
# Plotting Median Price per Bed in Quebec

fig = plt.figure(dpi=100)
ax = fig.add_axes([0,0,2,1])
ax.bar(df_qc_agg.UNIVERSITY,df_qc_agg.AVG_PRICE)
plt.xticks(rotation=90)
plt.title('Median Price per Bed - Quebec')
plt.show()
```



In [0]:

```
df_qc_beds = df_qc.groupby(['PROVINCE','UNIVERSITY'],as_index=False).agg({'BEDS': 'count'})
```

In [0]:

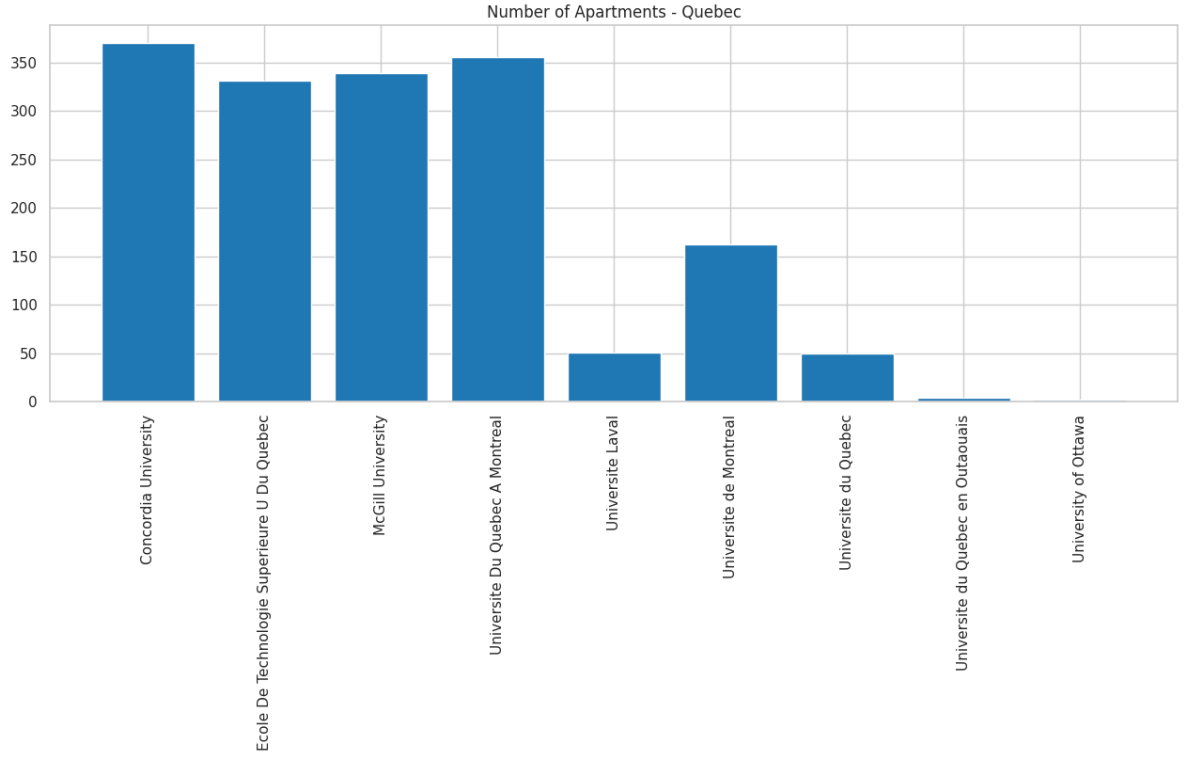
```
df_qc_beds.head(n=11)
```

Out[0]:

	PROVINCE	UNIVERSITY	BEDS
0	QC	Concordia University	370
1	QC	Ecole De Technologie Superieure U Du Quebec	331
2	QC	McGill University	339
3	QC	Universite Du Quebec A Montreal	356
4	QC	Universite Laval	51
5	QC	Universite de Montreal	162
6	QC	Universite du Quebec	50
7	QC	Universite du Quebec en Outaouais	4
8	QC	University of Ottawa	2

In [0]:

```
# Plotting Number of Available Apartments in Quebec

fig = plt.figure(dpi=100)
ax = fig.add_axes([0,0,2,1])
ax.bar(df_qc_beds.UNIVERSITY,df_qc_beds.BEDS)
plt.xticks(rotation=90)
plt.title('Number of Apartments - Quebec')
plt.show()
```



In [0]:

```
df_qc_beds = df_qc_beds.rename(columns={'UNIVERSITY': 'University'})
new_table_qc = pd.merge(df_qc_beds, QC_int_dom, on='University', how='inner')
new_table_qc.head(n=10)
```

Out[0]:

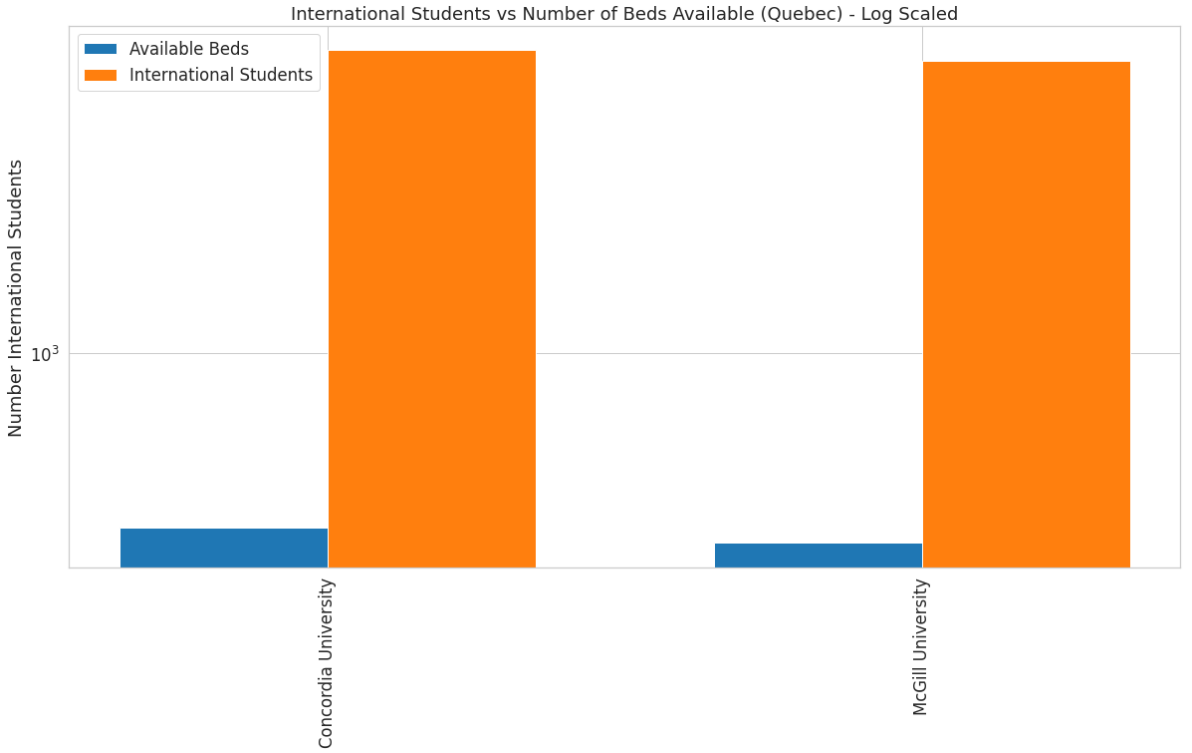
	PROVINCE	University	BEDS	Province	Total	International	Don
0	QC	Concordia University	370	Québec	39274	5618.371133	33655.6
1	QC	McGill University	339	Québec	36923	5282.047088	31640.9

In [0]:

```
# Plotting Quebec International Students vs Number of Beds Available

x = np.arange(len(new_table_qc.University))  # the label locations
width = 0.35  # the width of the bars
labels = new_table_qc.University
fig, ax = plt.subplots()
rects1 = ax.bar(x - width/2, new_table_qc.BEDS, width, label='Available Beds')
rects2 = ax.bar(x + width/2, new_table_qc.International, width, label='International Students')
fig.set_size_inches(20,10)
ax.set_yscale('log')
ax.set_ylabel('Number International Students')
ax.set_title('International Students vs Number of Beds Available (Quebec) - Log Scaled')
ax.set_xticks(x)
ax.set_xticklabels(labels)
plt.xticks(rotation=90)
ax.legend()
```

Out[0]:

```
<matplotlib.legend.Legend at 0x7fda6cc8a5c0>
```



In [0]:

```
#Objective 4: Determining Top 5 with the greatest gap

#Appending all the "new_table_province" tables

new_table_all = new_table_ab.append(new_table_bc, ignore_index = True).append(new_table_on, ignore_index = True).append(new_table_qc, ignore_index = True)
new_table_all['International'] = round(new_table_all['International'])

#Subseting this take to find the gaps - arranging in descending order
gap = new_table_all >> select(X.PROVINCE, X.University, X.BEDS, X.International) >> mutate(Gap = X.International-X.BEDS) >> arrange(X.Gap,ascending=False)
#gap

top_5 = gap.head(5)
top_5
```

Out[0]:

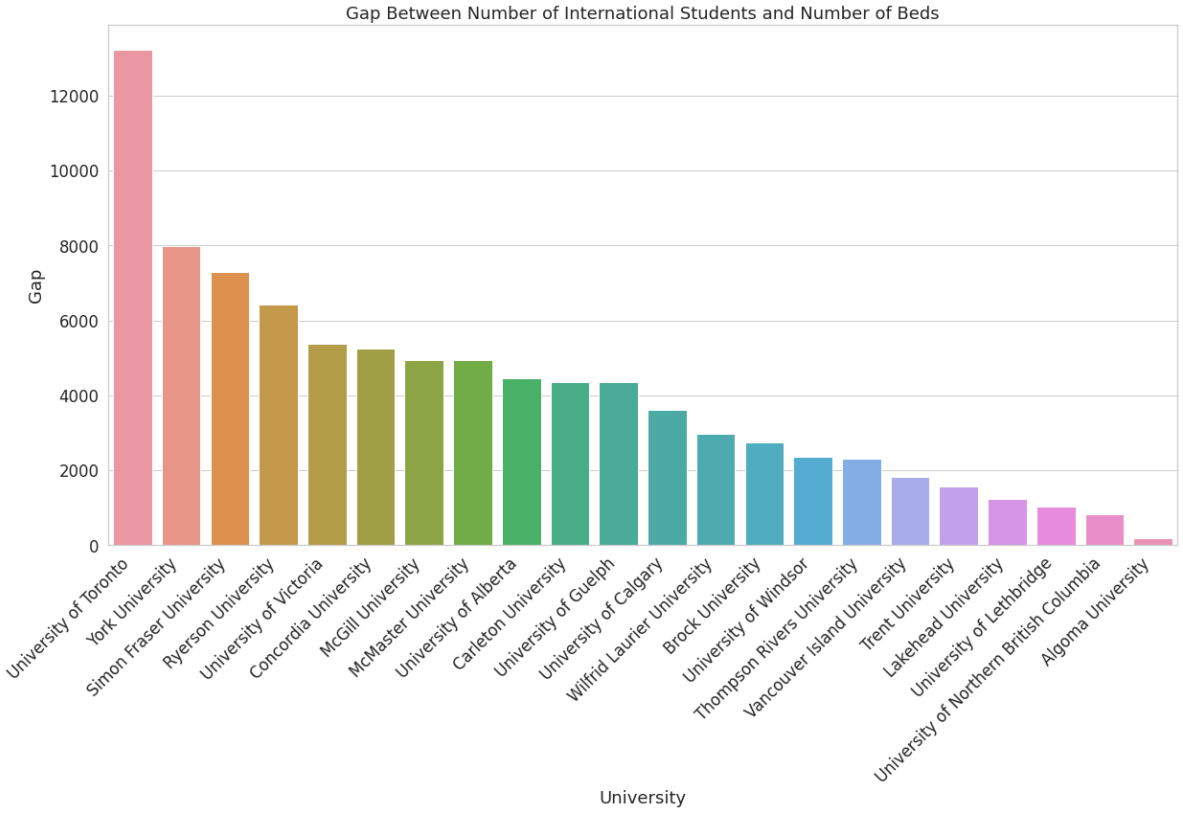
	PROVINCE	University	BEDS	International	Gap
16	ON	University of Toronto	194	13416.0	13222.0
19	ON	York University	123	8108.0	7985.0
3	BC	Simon Fraser University	60	7363.0	7303.0
13	ON	Ryerson University	478	6905.0	6427.0
6	BC	University of Victoria	15	5387.0	5372.0

In [0]:

```
#Plotting the gap

sns.set_context("notebook", font_scale=1.5, rc={"lines.linewidth
": 1.5})
sns.set_style("whitegrid")
plt.figure(figsize=(20, 10))

plt.title("Gap Between Number of International Students and Numb
er of Beds")
plt.xticks(rotation=45, horizontalalignment='right')

sns.barplot("University", "Gap", data=gap)
```

Out[0]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fda6d9b
5c50>
```



In [0]: