Characteristic	Lower, N = 2,358[1]	Upper, N = 2,066[1]
X	2,194 (1,086, 3,296)	2,224 (1,129, 3,341)
previous_qualification		
1	2,100 (89%)	1,617 (78%)
2	12 (0.5%)	11 (0.5%)
3	27 (1.1%)	99 (4.8%)
4	4 (0.2%)	4 (0.2%)
5	0 (0%)	1 (<0.1%)
6	11 (0.5%)	5 (0.2%)
9	10 (0.4%)	1 (<0.1%)
10	2 (<0.1%)	2 (<0.1%)
12	24 (1.0%)	21 (1.0%)
14	1 (<0.1%)	0 (0%)
15	2 (<0.1%)	0 (0%)
19	110 (4.7%)	52 (2.5%)
38	4 (0.2%)	3 (0.1%)
39	36 (1.5%)	183 (8.9%)
40	12 (0.5%)	28 (1.4%)
42	3 (0.1%)	33 (1.6%)
43	0 (0%)	6 (0.3%)
previous_qualification_grade_	127 (120, 133)	140 (132, 147)
mother_s_qualification		
1	563 (24%)	506 (24%)
2	39 (1.7%)	44 (2.1%)

[1] Median (IQR); n (%)

Characteristic	**Lower**, N = 2,358[1]	**Upper**, N = 2,066[1]
3	211 (8.9%)	227 (11%)
4	27 (1.1%)	22 (1.1%)
5	5 (0.2%)	16 (0.8%)
6	2 (<0.1%)	2 (<0.1%)
9	3 (0.1%)	5 (0.2%)
10	1 (<0.1%)	2 (<0.1%)
11	2 (<0.1%)	1 (<0.1%)
12	29 (1.2%)	13 (0.6%)
14	1 (<0.1%)	1 (<0.1%)
18	0 (0%)	1 (<0.1%)
19	525 (22%)	428 (21%)
22	0 (0%)	1 (<0.1%)
26	0 (0%)	1 (<0.1%)
27	1 (<0.1%)	0 (0%)
29	1 (<0.1%)	2 (<0.1%)
30	2 (<0.1%)	1 (<0.1%)
34	75 (3.2%)	55 (2.7%)
35	1 (<0.1%)	2 (<0.1%)
36	1 (<0.1%)	2 (<0.1%)
37	545 (23%)	464 (22%)
38	311 (13%)	251 (12%)
39	1 (<0.1%)	7 (0.3%)
40	1 (<0.1%)	8 (0.4%)
41	5 (0.2%)	1 (<0.1%)

[1] Median (IQR); n (%)

Characteristic	**Lower**, N = 2,358[1]	**Upper**, N = 2,066[1]
42	2 (<0.1%)	2 (<0.1%)
43	3 (0.1%)	1 (<0.1%)
44	1 (<0.1%)	0 (0%)
father_s_qualification		
1	464 (20%)	440 (21%)
2	34 (1.4%)	34 (1.6%)
3	120 (5.1%)	162 (7.8%)
4	15 (0.6%)	24 (1.2%)
5	12 (0.5%)	6 (0.3%)
6	2 (<0.1%)	0 (0%)
9	2 (<0.1%)	3 (0.1%)
10	0 (0%)	2 (<0.1%)
11	4 (0.2%)	6 (0.3%)
12	22 (0.9%)	16 (0.8%)
13	1 (<0.1%)	0 (0%)
14	2 (<0.1%)	2 (<0.1%)
18	0 (0%)	1 (<0.1%)
19	552 (23%)	416 (20%)
20	0 (0%)	1 (<0.1%)
22	3 (0.1%)	1 (<0.1%)
25	0 (0%)	1 (<0.1%)
26	0 (0%)	2 (<0.1%)
27	0 (0%)	1 (<0.1%)
29	3 (0.1%)	0 (0%)

[1] Median (IQR); n (%)

Characteristic	**Lower**, N = 2,358[1]	**Upper**, N = 2,066[1]
30	2 (<0.1%)	2 (<0.1%)
31	1 (<0.1%)	0 (0%)
33	0 (0%)	1 (<0.1%)
34	69 (2.9%)	43 (2.1%)
35	0 (0%)	2 (<0.1%)
36	3 (0.1%)	5 (0.2%)
37	660 (28%)	549 (27%)
38	377 (16%)	325 (16%)
39	6 (0.3%)	14 (0.7%)
40	2 (<0.1%)	3 (0.1%)
41	0 (0%)	2 (<0.1%)
42	1 (<0.1%)	0 (0%)
43	0 (0%)	2 (<0.1%)
44	1 (<0.1%)	0 (0%)
mother_s_occupation		
0	82 (3.5%)	62 (3.0%)
1	42 (1.8%)	60 (2.9%)
2	152 (6.4%)	166 (8.0%)
3	199 (8.4%)	152 (7.4%)
4	415 (18%)	402 (19%)
5	300 (13%)	230 (11%)
6	42 (1.8%)	49 (2.4%)
7	163 (6.9%)	109 (5.3%)
8	20 (0.8%)	16 (0.8%)

[1] Median (IQR); n (%)

Characteristic	**Lower**, N = 2,358[1]	**Upper**, N = 2,066[1]
9	835 (35%)	742 (36%)
10	2 (<0.1%)	2 (<0.1%)
90	42 (1.8%)	28 (1.4%)
99	11 (0.5%)	6 (0.3%)
122	1 (<0.1%)	1 (<0.1%)
123	4 (0.2%)	3 (0.1%)
125	0 (0%)	1 (<0.1%)
131	1 (<0.1%)	0 (0%)
132	2 (<0.1%)	1 (<0.1%)
134	1 (<0.1%)	3 (0.1%)
141	4 (0.2%)	4 (0.2%)
143	0 (0%)	3 (0.1%)
144	5 (0.2%)	1 (<0.1%)
151	0 (0%)	3 (0.1%)
152	1 (<0.1%)	1 (<0.1%)
153	1 (<0.1%)	1 (<0.1%)
171	1 (<0.1%)	0 (0%)
173	1 (<0.1%)	0 (0%)
175	3 (0.1%)	2 (<0.1%)
191	18 (0.8%)	8 (0.4%)
192	2 (<0.1%)	3 (0.1%)
193	3 (0.1%)	1 (<0.1%)
194	5 (0.2%)	6 (0.3%)
father_s_occupation		

[1] Median (IQR); n (%)

Characteristic	**Lower**, N = 2,358[1]	**Upper**, N = 2,066[1]
0	71 (3.0%)	57 (2.8%)
1	65 (2.8%)	69 (3.3%)
2	82 (3.5%)	115 (5.6%)
3	204 (8.7%)	180 (8.7%)
4	190 (8.1%)	196 (9.5%)
5	279 (12%)	237 (11%)
6	114 (4.8%)	128 (6.2%)
7	385 (16%)	281 (14%)
8	162 (6.9%)	156 (7.6%)
9	551 (23%)	459 (22%)
10	149 (6.3%)	117 (5.7%)
90	40 (1.7%)	25 (1.2%)
99	14 (0.6%)	5 (0.2%)
101	1 (<0.1%)	0 (0%)
102	0 (0%)	2 (<0.1%)
103	3 (0.1%)	1 (<0.1%)
112	0 (0%)	2 (<0.1%)
114	0 (0%)	1 (<0.1%)
121	1 (<0.1%)	0 (0%)
122	1 (<0.1%)	1 (<0.1%)
123	3 (0.1%)	0 (0%)
124	0 (0%)	1 (<0.1%)
131	0 (0%)	1 (<0.1%)
132	1 (<0.1%)	0 (0%)

[1] Median (IQR); n (%)

Characteristic	**Lower**, N = 2,358[1]	**Upper**, N = 2,066[1]
134	0 (0%)	1 (<0.1%)
135	1 (<0.1%)	2 (<0.1%)
141	0 (0%)	1 (<0.1%)
143	1 (<0.1%)	0 (0%)
144	5 (0.2%)	3 (0.1%)
151	0 (0%)	2 (<0.1%)
152	1 (<0.1%)	2 (<0.1%)
153	1 (<0.1%)	0 (0%)
154	1 (<0.1%)	0 (0%)
161	0 (0%)	1 (<0.1%)
163	3 (0.1%)	2 (<0.1%)
171	6 (0.3%)	2 (<0.1%)
172	1 (<0.1%)	1 (<0.1%)
174	1 (<0.1%)	0 (0%)
175	3 (0.1%)	1 (<0.1%)
181	3 (0.1%)	0 (0%)
182	2 (<0.1%)	0 (0%)
183	2 (<0.1%)	1 (<0.1%)
192	3 (0.1%)	3 (0.1%)
193	6 (0.3%)	9 (0.4%)
194	1 (<0.1%)	1 (<0.1%)
195	1 (<0.1%)	0 (0%)
admission_grade	118 (112, 122)	136 (130, 144)
educational_special_needs		

[1] Median (IQR); n (%)

Characteristic	**Lower**, N = 2,358[1]	**Upper**, N = 2,066[1]
0	2,324 (99%)	2,049 (99%)
1	34 (1.4%)	17 (0.8%)
gender		
0	1,539 (65%)	1,329 (64%)
1	819 (35%)	737 (36%)
age_at_enrollment	20 (19, 25)	20 (18, 25)
international		
0	2,308 (98%)	2,006 (97%)
1	50 (2.1%)	60 (2.9%)
unemployment_rate	11.10 (9.40, 12.70)	11.10 (9.40, 13.90)
inflation_rate		
-0.8	272 (12%)	261 (13%)
-0.3	186 (7.9%)	204 (9.9%)
0.3	183 (7.8%)	179 (8.7%)
0.5	241 (10%)	204 (9.9%)
0.6	259 (11%)	155 (7.5%)
1.4	478 (20%)	415 (20%)
2.6	306 (13%)	265 (13%)
2.8	220 (9.3%)	177 (8.6%)
3.7	213 (9.0%)	206 (10.0%)
gdp	0.32 (-1.70, 1.79)	0.32 (-1.70, 1.78)
target		
Dropout	809 (34%)	612 (30%)
Enrolled	471 (20%)	323 (16%)

[1] Median (IQR); n (%)

Characteristic	**Lower**, N = 2,358[1]	**Upper**, N = 2,066[1]
Graduate	1,078 (46%)	1,131 (55%)

[1] Median (IQR); n (%)

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

In [5]:
```python
df = pd.read_csv("graduation.csv", index_col=0)
```

In [3]:
```python
df.describe()
```

Out[3]:

	Unnamed: 0	Marital.status	Application.mode	Application.order	Course	Daytime.evening.attendance.
count	4424.000000	4424.000000	4424.000000	4424.000000	4424.000000	4424.000000
mean	2212.500000	1.178571	18.669078	1.727848	8856.642631	0.890823
std	1277.243125	0.605747	17.484682	1.313793	2063.566416	0.311897
min	1.000000	1.000000	1.000000	0.000000	33.000000	0.000000
25%	1106.750000	1.000000	1.000000	1.000000	9085.000000	1.000000
50%	2212.500000	1.000000	17.000000	1.000000	9238.000000	1.000000
75%	3318.250000	1.000000	39.000000	2.000000	9556.000000	1.000000
max	4424.000000	6.000000	57.000000	9.000000	9991.000000	1.000000

8 rows × 37 columns

In [4]:
```python
df.columns
```

Out[4]:
```
Index(['Unnamed: 0', 'Marital.status', 'Application.mode', 'Application.order',
       'Course', 'Daytime.evening.attendance.', 'Previous.qualification',
       'Previous.qualification..grade.', 'Nacionality',
       'Mother.s.qualification', 'Father.s.qualification',
       'Mother.s.occupation', 'Father.s.occupation', 'Admission.grade',
       'Displaced', 'Educational.special.needs', 'Debtor',
       'Tuition.fees.up.to.date', 'Gender', 'Scholarship.holder',
       'Age.at.enrollment', 'International',
       'Curricular.units.1st.sem..credited.',
       'Curricular.units.1st.sem..enrolled.',
       'Curricular.units.1st.sem..evaluations.',
       'Curricular.units.1st.sem..approved.',
       'Curricular.units.1st.sem..grade.',
       'Curricular.units.1st.sem..without.evaluations.',
       'Curricular.units.2nd.sem..credited.',
       'Curricular.units.2nd.sem..enrolled.',
       'Curricular.units.2nd.sem..evaluations.',
       'Curricular.units.2nd.sem..approved.',
       'Curricular.units.2nd.sem..grade.',
       'Curricular.units.2nd.sem..without.evaluations.', 'Unemployment.rate',
       'Inflation.rate', 'GDP', 'Target'],
      dtype='object')
```

Run t-tests against admission grade to check if these are confounders

Marital Status Daytime Evening Attendance GDP Inflation Rate Unemployment Rate Debtor

Drop:

Appl Mode Appl Order Course Nationality Displaced Tuition Scholarship

In [17]:
```python
df_confounders = df[
    [
```

```
        "Previous.qualification",
        "Previous.qualification..grade.",
        "Mother.s.qualification",
        "Father.s.qualification",
        "Mother.s.occupation",
        "Father.s.occupation",
        "Admission.grade",
        "Educational.special.needs",
        "Gender",
        "Age.at.enrollment",
        "International",
        "Unemployment.rate",
        "Inflation.rate",
        "GDP",
        "Target"
    ]
]
```

In [18]:

```
df_confounders.columns = list(map(lambda c: c.lower().replace(".", "_").replace("__", "_
df_confounders
```

Out[18]:

	previous_qualification	previous_qualification_grade_	mother_s_qualification	father_s_qualification	mother_s_(
1	1	122.0	19	12	
2	1	160.0	1	3	
3	1	122.0	37	37	
4	1	122.0	38	37	
5	1	100.0	37	38	
...	...	...	...	...	
4420	1	125.0	1	1	
4421	1	120.0	1	1	
4422	1	154.0	37	37	
4423	1	180.0	37	37	
4424	1	152.0	38	37	

4424 rows × 15 columns

In [19]:

```
df_confounders.to_csv("dat.csv")
```

In [10]:

```
df["GDP"].describe()
```

Out[10]:

```
count    4424.000000
mean        0.001969
std         2.269935
min        -4.060000
25%        -1.700000
50%         0.320000
75%         1.790000
max         3.510000
Name: GDP, dtype: float64
```

In [12]:

```
df["Debtor"].value_counts()
```

Out[12]:

```
0    3921
1     503
Name: Debtor, dtype: int64
```

In [11]:
```
df["Nacionality"].value_counts()
```

Out[11]:
```
1      4314
41       38
26       14
22       13
6        13
24        5
100       3
11        3
103       3
21        2
101       2
62        2
25        2
2         2
105       2
32        1
13        1
109       1
108       1
14        1
17        1
Name: Nacionality, dtype: int64
```

In [8]:
```
len(df.columns)
```

Out[8]:
```
38
```

In [9]:
```
df.shape
```

Out[9]:
```
(4424, 38)
```

In [21]:
```
df[['Curricular.units.1st.sem..credited.',
       'Curricular.units.1st.sem..enrolled.',
       'Curricular.units.1st.sem..evaluations.',
       'Curricular.units.1st.sem..approved.',
       'Curricular.units.1st.sem..grade.',
       'Curricular.units.1st.sem..without.evaluations.', 'Target']]
```

Out[21]:

	Curricular.units.1st.sem..credited.	Curricular.units.1st.sem..enrolled.	Curricular.units.1st.sem..evaluations.	Curri
0	0	0	0	
1	0	6	6	
2	0	6	0	
3	0	6	8	
4	0	6	9	
...	...	...	...	
4419	0	6	7	
4420	0	6	6	
4421	0	7	8	
4422	0	5	5	
4423	0	6	8	

4424 rows × 7 columns

In [5]:
```
df["Nacionality"].value_counts()
```
Out[5]:
```
1      4314
41       38
26       14
22       13
6        13
24        5
100       3
11        3
103       3
21        2
101       2
62        2
25        2
2         2
105       2
32        1
13        1
109       1
108       1
14        1
17        1
Name: Nacionality, dtype: int64
```
In [25]:
```
df["Course"].value_counts()
```
Out[25]:
```
12    766
9     380
10    355
6     337
15    331
14    268
17    268
11    252
5     226
2     215
3     215
4     210
16    192
7     170
8     141
13     86
1      12
Name: Course, dtype: int64
```
In [19]:
```
df.groupby("Target")["Curricular.units.1st.sem..credited."].mean()
```
Out[19]:
```
Target
Dropout     0.609430
Enrolled    0.507557
Graduate    0.847442
Name: Curricular.units.1st.sem..credited., dtype: float64
```
In [13]:
```
df["Target"].value_counts()
```
Out[13]:
```
Graduate    2209
Dropout     1421
Enrolled     794
Name: Target, dtype: int64
```
In [22]:
```
g = df.groupby("Target")
gg = g.get_group("Graduate")
gd = g.get_group("Dropout")
ge = g.get_group("Enrolled")
```

In [23]:
```
import scipy.stats as stats
# stats f_oneway functions takes the groups as input and returns ANOVA F and p value
fvalue, pvalue = stats.f_oneway(gg['Admission.grade'], gd['Admission.grade'], ge['Admiss
print(fvalue, pvalue)
```

```
35.64860425750162 4.380466113389808e-16
```

In [ ]:

In [2]:
```
# Loading the dataset and changing categorical variables to factors
df = read.csv("dat.csv")
df["previous_qualification"] = as.factor(df$previous_qualification)
df["mother_s_qualification"] = as.factor(df$mother_s_qualification)
df["father_s_qualification"] = as.factor(df$father_s_qualification)
df["mother_s_occupation"] = as.factor(df$mother_s_occupation)
df["father_s_occupation"] = as.factor(df$father_s_occupation)
df["educational_special_needs"] = as.factor(df$educational_special_needs)
df["gender"] = as.factor(df$gender)
df["international"] = as.factor(df$international)
df["target"] = as.factor(df$target)
head(df)
```

	X	previous_qualification	previous_qualification_grade_	mother_s_qualification	father_s_qualification	mothe
	<int>	<fct>	<dbl>	<fct>	<fct>	
1	1	1	122.0	19	12	
2	2	1	160.0	1	3	
3	3	1	122.0	37	37	
4	4	1	122.0	38	37	
5	5	1	100.0	37	38	
6	6	19	133.1	37	37	

In [19]:
```
# Loading libraries
library(twangContinuous)
library(cobalt)
library(survey)
library(gtsummary)
library(dplyr)
```

In [4]:
```
# Summary of the dataset
summary(df)
```

```
       X         previous_qualification previous_qualification_grade_
 Min.   :   1   1      :3717           Min.   : 95.0
 1st Qu.:1107   39     : 219           1st Qu.:125.0
 Median :2212   19     : 162           Median :133.1
 Mean   :2212   3      : 126           Mean   :132.6
 3rd Qu.:3318   12     :  45           3rd Qu.:140.0
 Max.   :4424   40     :  40           Max.   :190.0
                (Other): 115
 mother_s_qualification father_s_qualification mother_s_occupation
 1      :1069           37     :1209           9      :1577
 37     :1009           19     : 968           4      : 817
 19     : 953           1      : 904           5      : 530
 38     : 562           38     : 702           3      : 351
 3      : 438           3      : 282           2      : 318
 34     : 130           34     : 112           7      : 272
 (Other): 263           (Other): 247           (Other): 559
 father_s_occupation admission_grade educational_special_needs gender
 9      :1010        Min.   : 95.0   0:4373                    0:2868
 7      : 666        1st Qu.:117.9   1:  51                    1:1556
 5      : 516        Median :126.1
 4      : 386        Mean   :127.0
 3      : 384        3rd Qu.:134.8
 8      : 318        Max.   :190.0
 (Other):1144
```

```
 age_at_enrollment international unemployment_rate inflation_rate
 Min.   :17.00     0:4314        Min.   : 7.60     Min.   :-0.800
 1st Qu.:19.00     1: 110        1st Qu.: 9.40     1st Qu.: 0.300
 Median :20.00                   Median :11.10     Median : 1.400
 Mean   :23.27                   Mean   :11.57     Mean   : 1.228
 3rd Qu.:25.00                   3rd Qu.:13.90     3rd Qu.: 2.600
 Max.   :70.00                   Max.   :16.20     Max.   : 3.700

      gdp                target
 Min.   :-4.060000   Dropout :1421
 1st Qu.:-1.700000   Enrolled: 794
 Median : 0.320000   Graduate:2209
 Mean   : 0.001969
 3rd Qu.: 1.790000
 Max.   : 3.510000
```

In [8]:

```
# Column names
colnames(df)
```

'X' · 'previous_qualification' · 'previous_qualification_grade_' · 'mother_s_qualification' · 'father_s_qualification' · 'mother_s_occupation' · 'father_s_occupation' · 'admission_grade' · 'educational_special_needs' · 'gender' · 'age_at_enrollment' · 'international' · 'unemployment_rate' · 'inflation_rate' · 'gdp' · 'target'

ps.cont is a way of getting propensity scores for a continuous treatment variables (admission_grade in our case). Propensity scores are a probability of how likely a particular observation is to have that value of the treatment (check this definition)

In [9]:

```
# Calculation of Propensity Scores to use as weights for our regression model
psc.out <- ps.cont(admission_grade ~ previous_qualification +
                   previous_qualification_grade_ +
                   mother_s_qualification +
                   father_s_qualification +
                   mother_s_occupation +
                   father_s_occupation +
                   educational_special_needs +
                   gender +
                   international
                   , data = df)
summary(psc.out)
```

A matrix: 2 × 6 of type dbl

	n	ess	max.wcor	mean.wcor	rms.wcor	iter
unw	4424	4424.000	0.5804442	0.02516172	0.05686047	NA
AAC	4424	3556.534	0.2525825	0.02312257	0.03952497	45

In [10]:

```
head(df$admission_grade)
```

127.3 · 142.5 · 124.8 · 119.6 · 141.5 · 114.8

`w` are the weights, and there is one for each observation of the data. We use this to perform the survey weighted glm.

In [11]:

```
head(psc.out$w)
```

1: 0.905988619974881 **2:** 0.638202184695573 **3:** 0.881982325873705 **4:** 0.854787073872639 **5:** 1.4351401335233 **6:** 0.891158171133763

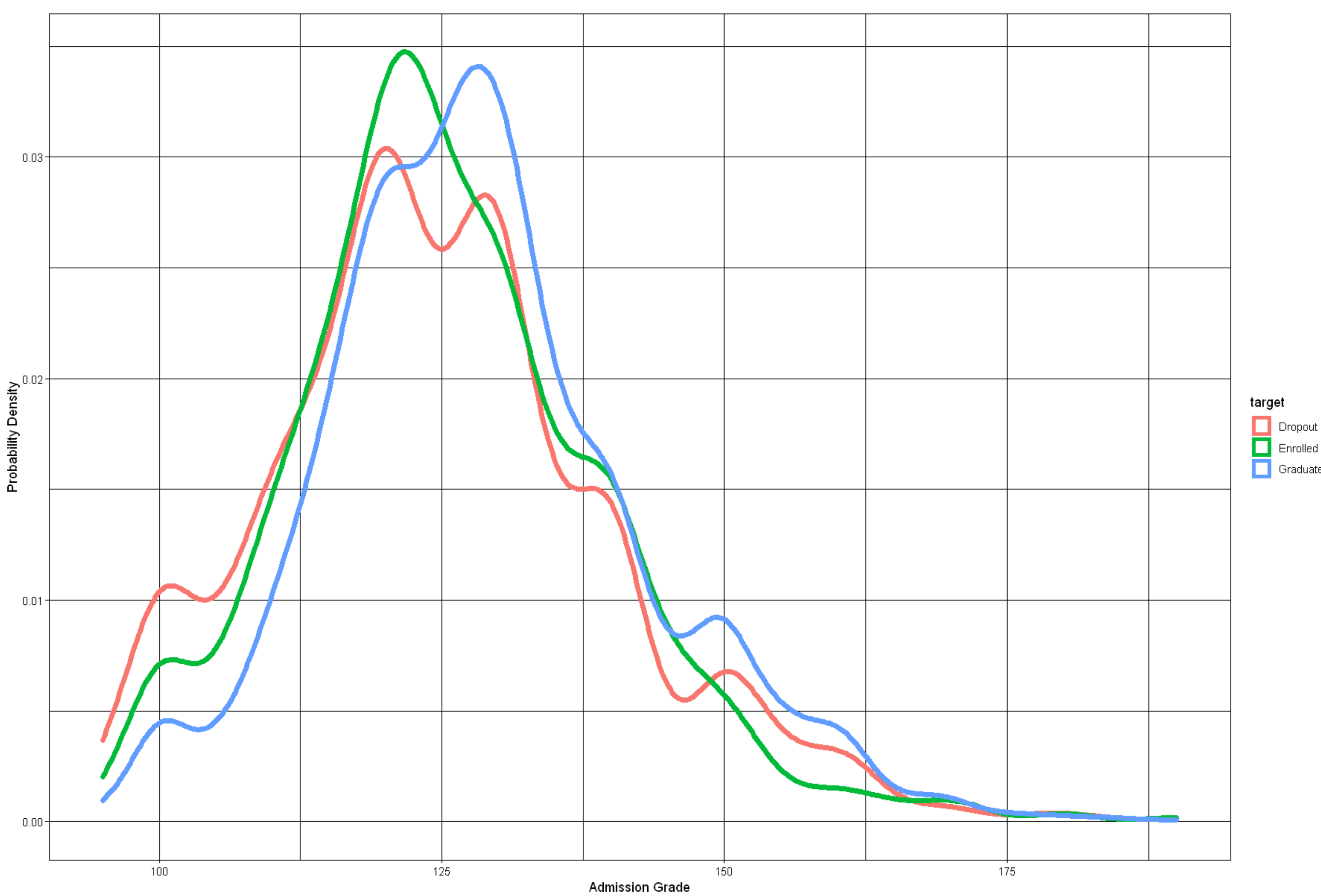
In [18]:
```
# Running the model
library(svyVGAM)
design <- svydesign(ids=~1, weights=psc.out$w, data=df)
mmodel <- svy_vglm(target ~ admission_grade, family=multinomial, design=design)
```

In [13]:
```
Z# Summary of the model
summary(mmodel)
```

```
svy_vglm.survey.design(target ~ admission_grade, family = multinomial,
    design = design)
Independent Sampling design (with replacement)
svydesign(ids = ~1, weights = psc.out$w, data = df)
                        Coef          SE        z           p
(Intercept):1      1.8325499  0.4079344  4.4923 7.047e-06
(Intercept):2      1.1172709  0.4504119  2.4806   0.01312
admission_grade:1 -0.0175567  0.0032498 -5.4024 6.574e-08
admission_grade:2 -0.0167687  0.0035593 -4.7113 2.462e-06
```

Interpretation: With 1 point increase in admission_grade, chance of dropout (`admission_grade:1`) goes down by 1.8% and chance of still being enrolled (failed to graduate in stipulated time) goes down by 1.7%

In [15]:
```
# Plotting the figure
library(ggplot2)
options(repr.plot.width = 15, repr.plot.height =10)
ggplot(df, aes(x = admission_grade)) +
  geom_density(aes(color = target), size=2) +
  xlab("Admission Grade") +
  ylab("Probability Density") + theme_linedraw() +
  labs("Density Plot for Admission Grade across Graduation Status")
```



In [16]:
```
# Descriptive statistics for our dataset using admission grade as two groups
data = df
data$group = with(df, ifelse(admission_grade > 127, 'Upper', 'Lower'))
glimpse(data)
```

```
Rows: 4,424
Columns: 17
$ X                              <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1…
$ previous_qualification         <fct> 1, 1, 1, 1, 1, 19, 1, 1, 1, 1, 1, 1, 1, …
$ previous_qualification_grade_  <dbl> 122.0, 160.0, 122.0, 122.0, 100.0, 133.1…
$ mother_s_qualification         <fct> 19, 1, 37, 38, 37, 37, 19, 37, 1, 1, 38,…
$ father_s_qualification         <fct> 12, 3, 37, 37, 38, 37, 38, 37, 1, 19, 19…
$ mother_s_occupation            <fct> 5, 3, 9, 5, 9, 9, 7, 9, 9, 4, 5, 9, 4, 4…
$ father_s_occupation            <fct> 9, 3, 9, 3, 9, 7, 10, 9, 9, 7, 7, 9, 9, …
$ admission_grade                <dbl> 127.3, 142.5, 124.8, 119.6, 141.5, 114.8…
$ educational_special_needs      <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
$ gender                         <fct> 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0…
$ age_at_enrollment              <int> 20, 19, 19, 20, 45, 50, 18, 22, 21, 18, …
$ international                  <fct> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0…
$ unemployment_rate              <dbl> 10.8, 13.9, 10.8, 9.4, 13.9, 16.2, 15.5,…
$ inflation_rate                 <dbl> 1.4, -0.3, 1.4, -0.8, -0.3, 0.3, 2.8, 2.…
$ gdp                            <dbl> 1.74, 0.79, 1.74, -3.12, 0.79, -0.92, -4…
$ target                         <fct> Dropout, Graduate, Dropout, Graduate, Gr…
$ group                          <chr> "Upper", "Upper", "Lower", "Lower", "Upp…
```

In [17]:

```
t = tbl_summary(data, by = group)
dat1 = data[c(3,11,13,14,15,17)]
t2 = dat1 %>%
  tbl_summary(by = group, type = list(where(is.numeric) ~ "continuous2")) %>%
  add_p(all_continuous() ~ "t.test")
```