Estimation Practicum

CAPM Replication

General Regression Techniques

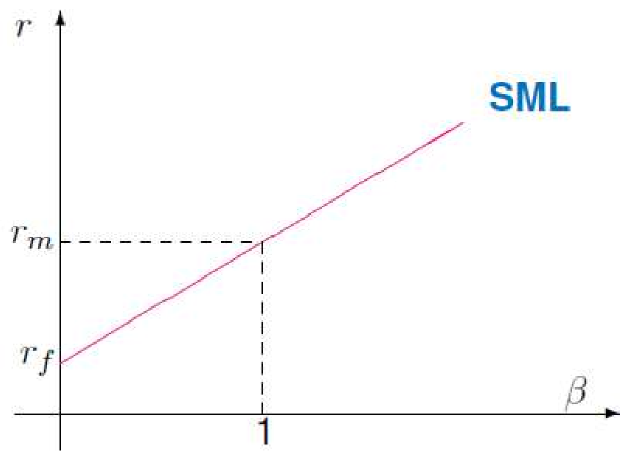
- Fixed Effects
- Dealing with outliers via ln()
- Clustering of standard errors
- Interaction and squared terms
- Collinearity
- Discrete dependent variables
- Probit, Logit, Multinomial and Ordered

Dataset and code same as for class 3. If you have not installed Python yet, make this a priority

CAPM Tests

Undervalued versus Overvalued

- From (5), we see that the market portfolio has $\beta_m = 1$.
- You can easily draw an SML with two points $(0, r_f)$ and $(1, r_m)$.



Implications and Tests of CAPM

- If **CAPM were correct**:
  - OLS regression of excess asset return against excess market returns for any asset **would have a constant term of** zero
  - OLS regression of excess asset return against excess market returns **+ {any other variable on the RHS}** would have a {~~significant~~ | insignificant} coefficient on the other variables
  - Above **two points help us to test CAPM empirically**
- Some **CAPM tests**:
  - Fama-French showed that in addition to market beta, company size (proxied by market cap) and valuation (book to market ratio) are also significant. This became **Fama-French "3 factor model"**
  - Carhart, proposed a 4^{th} factor, **which is momentum, proxied by lagged returns of the stock**
  - Thereafter, the 'flood gates' opened, and **researchers started adding and testing additions using OLS framework**

Table 1
Factor classification

Risk type		Description	Examples
Common (113)	**Financial** (46)	Proxy for aggregate financial market movement, including market portfolio returns, volatility, squared market returns, among others	Sharpe (1964): market returns; Kraus and Litzenberger (1976): squared market returns
	Macro (40)	Proxy for movement in macroeconomic fundamentals, including consumption, investment, inflation, among others	Breeden (1979): consumption growth; Cochrane (1991): investment returns
	Microstructure (11)	Proxy for aggregate movements in market microstructure or financial market frictions, including liquidity, transaction costs, among others	Pastor and Stambaugh (2003): market liquidity; Lo and Wang (2006): market trading volume
	Behavioral (3)	Proxy for aggregate movements in investor behavior, sentiment or behavior-driven systematic mispricing	Baker and Wurgler (2006): investor sentiment; Hirshleifer and Jiang (2010): market mispricing
	Accounting (8)	Proxy for aggregate movement in firm-level accounting variables, including payout yield, cash flow, among others	Fama and French (1992): size and book-to-market; Da and Warachka (2009): cash flow
	Other (5)	Proxy for aggregate movements that do not fall into the above categories, including momentum, investors' beliefs, among others	Carhart (1997): return momentum; Ozoguz (2009): investors' beliefs
Characteristics (202)	**Financial** (61)	Proxy for firm-level idiosyncratic financial risks, including volatility, extreme returns, among others	Ang et al. (2006): idiosyncratic volatility; Bali, Cakici, and Whitelaw (2011): extreme stock returns
	Microstructure (28)	Proxy for firm-level financial market frictions, including short sale restrictions, transaction costs, among others	Jarrow (1980): short sale restrictions; Mayshar (1981): transaction costs
	Behavioral (3)	Proxy for firm-level behavioral biases, including analyst dispersion, media coverage, among others	Diether, Malloy, and Scherbina (2002): analyst dispersion; Fang and Peress (2009): media coverage
	Accounting (87)	Proxy for firm-level accounting variables, including PE ratio, debt-to-equity ratio, among others	Basu (1977): PE ratio; Bhandari (1988): debt-to-equity ratio
	Other (24)	Proxy for firm-level variables that do not fall into the above categories, including political campaign contributions, ranking-related firm intangibles, among others	Cooper, Gulen, and Ovtchinnikov (2010): political campaign contributions; Edmans (2011): intangibles

The numbers in parentheses represent the number of factors identified. See Table 6 and http://faculty.fuqua.duke.edu/~charvey/Factor-List.xlsx.

Currently, there are **around 300 factors** published in top tier academic journals:
https://academic.oup.com/rfs/article/29/1/5/1843824

Application of CAPM: Cross Sectional Stock Picking

- Ideal stock to hold:
  - positive alpha
  - large beta during bull market
  - small beta during bear market
  - large Sharpe ratio
- Ideal stock to short is the reverse.
- Long the “good” stocks, short the “bad” stocks. Will this quant strategy work?
- So exactly how could one search for those good and bad stocks?

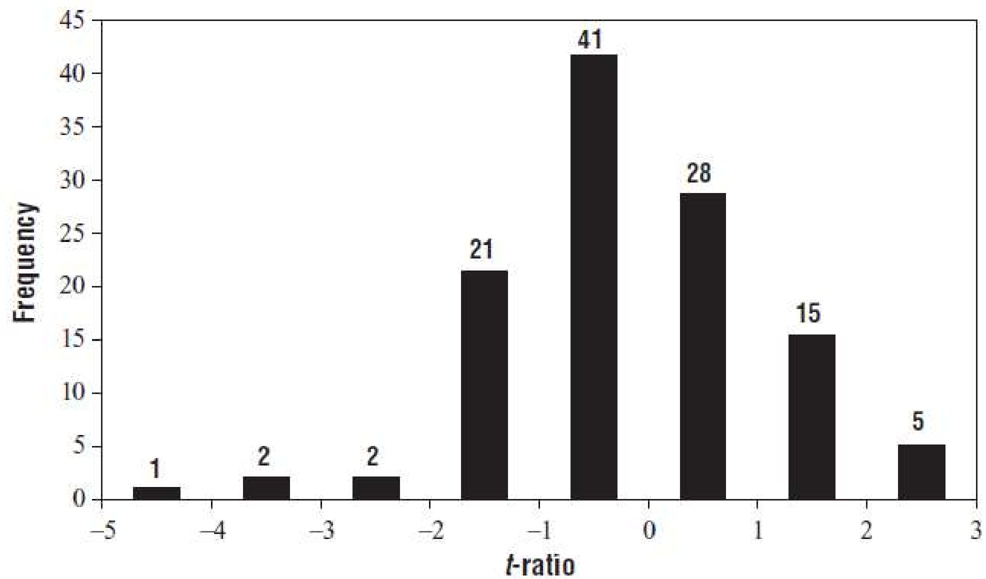
Compare Cross Sectional Stock Picking with Market Timing

- Shift investment funds into the market portfolio when market is rising, and to shift out of the stock market into money market when market is falling, especially if the market falls below risk-free return.
- The goal is to avoid being invested in e.g. mutual funds during a market decline.
- Typically, trend-following indicators are used to determine the direction and identify buy and sell signals.
- In an up move “buy signal,” money is transferred from a money market fund into a mutual fund in an attempt to capture a capital gain.
- In a down move “sell signal,” the assets in the mutual fund are sold and moved back into the money market for safe keeping until the next up move.

Jensen's Empirical Tests

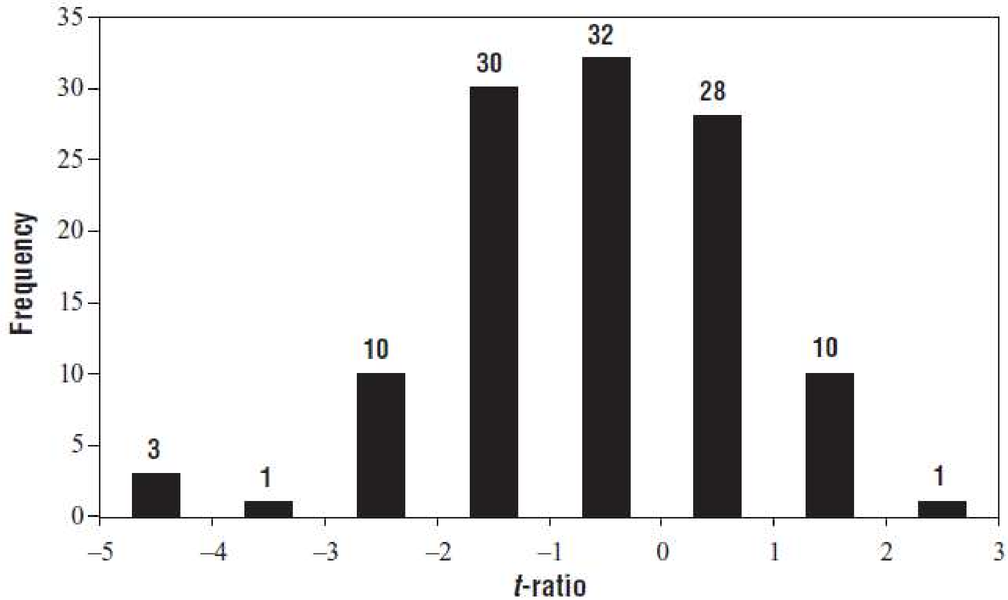
- Testing for the presence and significance of abnormal returns ("Jensen's alpha" - Jensen, 1968).
- Data: Annual Returns on the portfolios of 115 mutual funds from 1945–1964.
- The model: $R_{jt} - R_{ft} = \alpha_j + \beta_j\,(R_{mt} - R_{ft}) + u_{jt}$ for $j = 1, 2, \ldots, 115$.
- Are α_j significant?
- The null hypothesis is H_0: $\alpha_j = 0$.

Distribution of Mutual Fund Alphas' t-Ratios



Jensen (1968). Reprinted with the permission of Blackwell publishers.

Net of Transactions Costs



Jensen (1968). Reprinted with the permission of Blackwell publishers.

General Regression Techniques

In this section, we implement some regressions in python using both OLS as well as logit models, as well as explore some practical issues with outliers, fixed effects, and standard error estimation

Python packages used are fully open source (pandas, numpy, statsmodels), and are also widely used in industry or research academia.

There are many python versions. One convenient version that will allow you to follow along is here (https://www.anaconda.com/products/individual). Notwithstanding, we make no judgement that any version of python is better or worse; if you already have it installed, just use pre-existing

Datasets and sample code

Same as for Class 3

- On E-Learn:
  - **corpfund.csv** contains panel data on corporate performance (e.g. EBITDA, net income, revenue, etc) for US listed firms
  - **stockmetadata.csv** contains company descriptions such as tickers, industry, geographical location, etc
  - **ols_tests.py** contains python code that ingests and merges the data, as well as runs statistical analysis
  - Do not share data with anyone who is not taking this class

Data ingestation and merging steps

```
1  import pandas as pd;
2  import numpy as np;
3  import statsmodels.api as sm;
4  import statsmodels.discrete.discrete_model as smdiscrete
5  pd.set_option('use_inf_as_na', True)
6
7  meta_df = pd.read_csv("stockmetadata.csv")
8  fdata_df = pd.read_csv("corpfund.csv")
9  fdata_df = fdata_df[fdata_df['dimension']=='ARQ']
10 fdata_df['datekey'] = pd.to_datetime(fdata_df['datekey'])
11 df_left = pd.merge(fdata_df, meta_df, on='ticker', how='left')
12 df_left = df_left.set_index('datekey')
```

1. Lines 7 & 8: read in both csv files as python dataframes
2. A python dataframe is a table with (usually) time on the x-axis and various variables in columns
3. In line 9, we only keep “as reported data” to avoid forward bias due to corporate restatements
4. In line 11, we merge both dataframes based on ticker
5. In line 12, we set index of the dataframe to be data-date

```
In [2]: df_left
Out[2]:
           ticker dimension calendardate reportperiod lastupdated_x      accoci  ... firstpricedate lastpricedate firstquarter lastquarter                                          secfilings                      companysite
datekey                                                                          ...
2010-06-07      A       ARQ   2010-03-31   2010-04-30    2020-09-01 -239000000.0  ...     1999-11-18    2020-10-09   1997-06-30  2020-06-30  https://www.sec.gov/cgi-bin/browse-edgar?actio...          http://www.agilent.com
2010-09-07      A       ARQ   2010-06-30   2010-07-31    2020-09-01 -225000000.0  ...     1999-11-18    2020-10-09   1997-06-30  2020-06-30  https://www.sec.gov/cgi-bin/browse-edgar?actio...          http://www.agilent.com
2010-12-20      A       ARQ   2010-09-30   2010-10-31    2020-09-01  -88000000.0  ...     1999-11-18    2020-10-09   1997-06-30  2020-06-30  https://www.sec.gov/cgi-bin/browse-edgar?actio...          http://www.agilent.com
2011-03-09      A       ARQ   2010-12-31   2011-01-31    2020-09-01  -63000000.0  ...     1999-11-18    2020-10-09   1997-06-30  2020-06-30  https://www.sec.gov/cgi-bin/browse-edgar?actio...          http://www.agilent.com
2011-06-07      A       ARQ   2011-03-31   2011-04-30    2020-09-01  278000000.0  ...     1999-11-18    2020-10-09   1997-06-30  2020-06-30  https://www.sec.gov/cgi-bin/browse-edgar?actio...          http://www.agilent.com
...           ...       ...          ...          ...           ...          ...  ...            ...           ...          ...         ...                                                ...                             ...
2020-10-05   STTK       ARQ   2019-12-31   2019-12-31    2020-10-09      54000.0  ...     2020-10-09    2020-10-09   2018-12-31  2020-06-30  https://www.sec.gov/cgi-bin/browse-edgar?actio...     http://www.shattucklabs.com
2020-10-08   STTK       ARQ   2020-06-30   2020-06-30    2020-10-09      18000.0  ...     2020-10-09    2020-10-09   2018-12-31  2020-06-30  https://www.sec.gov/cgi-bin/browse-edgar?actio...     http://www.shattucklabs.com
2020-10-08   KRON       ARQ   2020-06-30   2020-06-30    2020-10-09     164000.0  ...     2020-10-09    2020-10-09   2018-12-31  2020-06-30  https://www.sec.gov/cgi-bin/browse-edgar?actio...          http://kronosbio.com
2020-09-18   SPRB       ARQ   2019-12-31   2019-12-31    2020-10-09          0.0  ...     2020-10-09    2020-10-09   2018-12-31  2020-06-30  https://www.sec.gov/cgi-bin/browse-edgar?actio...  http://www.sprucebiosciences.com
2020-10-05   SPRB       ARQ   2020-06-30   2020-06-30    2020-10-09          0.0  ...     2020-10-09    2020-10-09   2018-12-31  2020-06-30  https://www.sec.gov/cgi-bin/browse-edgar?actio...  http://www.sprucebiosciences.com

[207494 rows x 138 columns]
```

This is what dataset looks like after running code from previous slide

Fixed effects

- We want to estimate relationship between E/P ratio and operating profit margin for each of these data point
- However, we note different industries may have completely different operating margins on average, as well as different relationships between operating margin and E/P ratio
- One solution is to introduce an indicator variable (“dummy variable”) for each industry
- This allows each industry to effectively have its own y-intercept

Fixed effects specification

OLS without fixed effects:

$$Y_i = \alpha + \beta X_i + \varepsilon$$

OLS with fixed effects:

$$Y_i = \alpha + \beta_1 X_i + \beta_2 * I_1 + \beta_3 * I_2 + \beta_4 * I_3 + \ldots \beta_{12} * I_{12} + \varepsilon$$

Where

I_1 = 1 if firm is in sector 1, 0 otherwise
I_2 = 1 if firm is in sector 2, 0 otherwise
…
I_{12} = 1 if firm is in sector 12, 0 otherwise

Consequently:

Effective y-intercept for datapoints in sector 1 = $\alpha + \beta_2$
Effective y-intercept for datapoints in sector 2 = $\alpha + \beta_3$, etc

```
15  industrydummies = pd.get_dummies(df_left['sicsector'])
16  industrydummies.sum()       #purely for exploring the data, has no other purpose
17  industrydummies.describe()      #purely for exploring the data, has no other purpose
18
19  data_w_dummies = pd.concat([df_left,industrydummies], axis=1)
20
21  data_w_dummies.drop(['Wholesale Trade'], inplace=True, axis=1)  #drop 1 dummy variable
22  data_w_dummies['epratio'] = data_w_dummies['eps']/data_w_dummies['price']   #generate dependent variable
23  data_w_dummies['operatingmargin'] = data_w_dummies['opinc'] / data_w_dummies['revenue']     #generate independent var
```

Fixed effects example code

- Line 15, we generate 1 dummy variable for each sector
- Line 19, we merge the dummy variable dataframe into the main dataframe
- Line 21, we drop 1 dummy variable, which will be the control group
- Line 22, we generate our dependent variable (E/P ratio)
- Line 23, we generate our independent variable, which is operating margin

```
In [3]: industrydummies
Out[3]:
            Agriculture Forestry And Fishing  Construction  Finance Insurance And Real Estate  Manufacturing  Mining  Retail Trade  Services  Transportation Communications Electric Gas And Sanitary Service  Wholesale Trade
datekey
2010-06-07                                 0             0                                  0              1       0             0         0                                                                0                0
2010-09-07                                 0             0                                  0              1       0             0         0                                                                0                0
2010-12-20                                 0             0                                  0              1       0             0         0                                                                0                0
2011-03-09                                 0             0                                  0              1       0             0         0                                                                0                0
2011-06-07                                 0             0                                  0              1       0             0         0                                                                0                0
...                                      ...           ...                                ...            ...     ...           ...       ...                                                              ...              ...
2020-10-05                                 0             0                                  0              1       0             0         0                                                                0                0
2020-10-08                                 0             0                                  0              1       0             0         0                                                                0                0
2020-10-08                                 0             0                                  0              1       0             0         0                                                                0                0
2020-09-18                                 0             0                                  0              1       0             0         0                                                                0                0
2020-10-05                                 0             0                                  0              1       0             0         0                                                                0                0

[207494 rows x 9 columns]
```

This is what ‘industrydummies’ dataframe looks like after line 15 in the previous slide

```python
26 #initial analysis
27 result = sm.OLS(data_w_dummies['epratio'], sm.add_constant(data_w_dummies[['operatingmargin']]), missing='drop').fit()
28 result.summary()
29 result = sm.OLS(data_w_dummies['epratio'], sm.add_constant(data_w_dummies[['operatingmargin', 'Agriculture Forestry And Fishing', 'Construction','Finance Insurance And Real Estate',
30 result.summary()
```

OLS regression with and without dummy variables

- Line 27 and 28, we run the OLS regression without dummy variables
- Line 29 and 30, we include the dummy variables

```
In [4]: result = sm.OLS(data_w_dummies['epratio'], sm.add_constant(data_w_dummies[['operatingmargin']]),
   ...: result.summary()
Out[4]:
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                epratio   R-squared:                       0.000
Model:                            OLS   Adj. R-squared:                 -0.000
Method:                 Least Squares   F-statistic:                 4.803e-06
Date:                Sat, 10 Oct 2020   Prob (F-statistic):              0.998
Time:                        16:58:52   Log-Likelihood:            -6.4512e+05
No. Observations:              115861   AIC:                         1.290e+06
Df Residuals:                  115859   BIC:                         1.290e+06
Df Model:                           1
Covariance Type:            nonrobust
===================================================================================
                      coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------
const               0.2554      0.186      1.372      0.170      -0.110       0.620
operatingmargin  4.402e-06      0.002      0.002      0.998      -0.004       0.004
==============================================================================
Omnibus:                   651923.203   Durbin-Watson:                   1.999
Prob(Omnibus):                  0.000   Jarque-Bera (JB):   60232700557916.930
Skew:                         331.783   Prob(JB):                         0.00
Kurtosis:                  111701.024   Cond. No.                         92.7
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

No Fixed Effects

```
   ...: result.summary()
Out[5]:
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                epratio   R-squared:                       0.000
Model:                            OLS   Adj. R-squared:                 -0.000
Method:                 Least Squares   F-statistic:                    0.3172
Date:                Sat, 10 Oct 2020   Prob (F-statistic):              0.970
Time:                        16:59:55   Log-Likelihood:            -6.4512e+05
No. Observations:              115861   AIC:                         1.290e+06
Df Residuals:                  115851   BIC:                         1.290e+06
Df Model:                           9
Covariance Type:            nonrobust
=====================================================================================================================================
                                                                 coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------------------------------------------------------
const                                                          0.7960      0.372      2.141      0.032       0.067       1.525
operatingmargin                                            -1.244e-05      0.002     -0.006      0.995      -0.004       0.004
Agriculture Forestry And Fishing                              -0.7124      3.933     -0.181      0.856      -8.422       6.997
Construction                                                  -0.7724      1.732     -0.446      0.656      -4.168       2.623
Finance Insurance And Real Estate                             -0.6747      0.531     -1.270      0.204      -1.716       0.366
Manufacturing                                                 -0.7171      0.525     -1.367      0.172      -1.745       0.311
Mining                                                        -0.7509      1.226     -0.612      0.540      -3.154       1.653
Retail Trade                                                  -0.7757      0.927     -0.837      0.402      -2.592       1.040
Services                                                      -0.7737      0.689     -1.122      0.262      -2.125       0.577
Transportation Communications Electric Gas And Sanitary Service   -0.7695      0.822     -0.936      0.349      -2.380       0.841
==============================================================================
Omnibus:                   651916.510   Durbin-Watson:                   1.999
Prob(Omnibus):                  0.000   Jarque-Bera (JB):   60226461523137.062
Skew:                         331.771   Prob(JB):                         0.00
Kurtosis:                  111695.238   Cond. No.                     1.96e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.96e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

With Fixed Effects

We note t-statistic for estimated coefficient on 'operatingmargin' is very low

This might be due to the effect of outliners overly influencing estimation

Two ways to explore the effect of outliers are:

- Winsorize data [albeit at expense of deleting observations arbitrarily ("datamining")]
- Apply logarithm to both dependent and independent variables. This will naturally reduce impact of outliers

```
31 data_w_dummies['lnoperatingmargin'] = np.log(data_w_dummies['operatingmargin'])
32 result = sm.OLS(data_w_dummies['epratio'], sm.add_constant(data_w_dummies[['lnoperatingmargin']]), missing='drop').fit()
33 result.summary()
34 data_w_dummies['lnepratio'] = np.log(data_w_dummies['epratio'])
35 result = sm.OLS(data_w_dummies['lnepratio'], sm.add_constant(data_w_dummies[['lnoperatingmargin']]), missing='drop').fit()
36 result.summary()
37
38 #with dummy variables
39 result = sm.OLS(data_w_dummies['lnepratio'], sm.add_constant(data_w_dummies[['lnoperatingmargin', 'Agriculture Forestry And Fishing', 'Construction','Finance Insurance And Real Estate'
40 result.summary()
```

- Lines 31 and 34: Generate log versions of both dependent and independent variables
- Line 35: Run regression with log variables
- Line 39: Same regression, but with dummy variables included

```
In [6]: result = sm.OLS(data_w_dummies['lnepratio'], sm.add_constant(data_w_dummies[['lnoperatingmargin']]), missing='drop').fit()^M
   ...: result.summary()^M
   ...:
Out[6]:
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:              lnepratio   R-squared:                       0.071
Model:                            OLS   Adj. R-squared:                  0.071
Method:                 Least Squares   F-statistic:                     8846.
Date:                Sat, 10 Oct 2020   Prob (F-statistic):               0.00
Time:                        17:08:40   Log-Likelihood:            -1.5681e+05
No. Observations:              115861   AIC:                         3.136e+05
Df Residuals:                  115859   BIC:                         3.136e+05
Df Model:                           1
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const                -3.7721      0.006   -602.657      0.000      -3.784      -3.760
lnoperatingmargin     0.2638      0.003     94.054      0.000       0.258       0.269
==============================================================================
Omnibus:                    18365.631   Durbin-Watson:                   1.036
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           158342.851
Skew:                           0.510   Prob(JB):                         0.00
Kurtosis:                       8.635   Cond. No.                         5.93
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
"""
```

Log variables, no FE

```
In [7]: result = sm.OLS(data_w_dummies['lnepratio'], sm.add_constant(data_w_dummies[['lnoperatingmargin', 'Agriculture Forestry And Fishi
   ...: 'Services','Transportation Communications Electric Gas And Sanitary Service']]), missing='drop').fit()^M
   ...: result.summary()
Out[7]:
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:              lnepratio   R-squared:                       0.085
Model:                            OLS   Adj. R-squared:                  0.084
Method:                 Least Squares   F-statistic:                     1188.
Date:                Sat, 10 Oct 2020   Prob (F-statistic):               0.00
Time:                        17:09:31   Log-Likelihood:            -1.5595e+05
No. Observations:              115861   AIC:                         3.119e+05
Df Residuals:                  115851   BIC:                         3.120e+05
Df Model:                           9
Covariance Type:            nonrobust
======================================================================================================================================
                                                                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------------------------------------------------------------
const                                                           -3.6332      0.009   -419.187      0.000      -3.650      -3.616
lnoperatingmargin                                                0.2969      0.003     96.466      0.000       0.291       0.303
Agriculture Forestry And Fishing                                 0.6587      0.058     11.417      0.000       0.546       0.772
Construction                                                     0.3404      0.025     13.352      0.000       0.290       0.390
Finance Insurance And Real Estate                               -0.1784      0.008    -21.688      0.000      -0.195      -0.162
Manufacturing                                                   -0.0182      0.008     -2.360      0.018      -0.033      -0.003
Mining                                                          -0.0450      0.018     -2.489      0.013      -0.080      -0.010
Retail Trade                                                     0.1256      0.014      9.187      0.000       0.099       0.152
Services                                                        -0.2630      0.010    -26.005      0.000      -0.283      -0.243
Transportation Communications Electric Gas And Sanitary Service -0.1285      0.012    -10.640      0.000      -0.152      -0.105
==============================================================================
Omnibus:                    18757.262   Durbin-Watson:                   1.031
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           159948.347
Skew:                           0.533   Prob(JB):                         0.00
Kurtosis:                       8.657   Cond. No.                         51.5
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
"""
```

Log variables, with FE

Clustering of Standard Errors

We note t-statistic is significantly improved, but could this be a result of intra-cluster correlations?

- Observations within same industry are likely to be highly correlated versus observations in different industries
- This violates OLS assumption of iid datapoints
- Estimates may therefore greatly overstate efficiency (inflated t-statistics, artificially low p-values, downward biases standard errors)
- We can adjust the standard error estimates by clustering errors within each industry
- Background: See for e.g. https://economics.mit.edu/files/13927 for more information

For next couple of slides, we focus on empirical application rather than theoretical proof of forgoing

```
42  #clustering standard variables
43  data_w_dummies.dropna(subset = ['lnepratio', 'lnoperatingmargin'], inplace=True)   #because of a bug in python where fillna is not working perfectly
44  #note that we can cannot cluster by str variables in python, hence using siccode instead of sicsector
45  result = sm.OLS(data_w_dummies['lnepratio'], sm.add_constant(data_w_dummies[['lnoperatingmargin', 'Agriculture Forestry And Fishing', 'Construction','Finance Insurance And Real Es
    'Trade', 'Services','Transportation Communications Electric Gas And Sanitary Service']]), missing='drop').fit(cov_type='cluster', cov_kwds={'groups': data_w_dummies['siccode']})
46  result.summary()
47
```

Line 43: We drop all observations with missing data to avoid a bug in python

Line 45: Run OLS estimation on log variables with FE, and standard errors clustered by sic_code

```
In [8]: result = sm.OLS(data_w_dummies['lnepratio'], sm.add_constant(data_w_dummies[['lnoperatingmargin', 'Agriculture Forestry And Fishir
   ...: 'Services','Transportation Communications Electric Gas And Sanitary Service']]), missing='drop').fit(cov_type='cluster', cov_kwds=
   ...: result.summary()
   ...:
Out[8]:
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:              lnepratio   R-squared:                       0.085
Model:                            OLS   Adj. R-squared:                  0.084
Method:                 Least Squares   F-statistic:                     44.65
Date:                Sat, 10 Oct 2020   Prob (F-statistic):           2.30e-80
Time:                        17:22:14   Log-Likelihood:            -1.5595e+05
No. Observations:              115861   AIC:                         3.119e+05
Df Residuals:                  115851   BIC:                         3.120e+05
Df Model:                           9
Covariance Type:              cluster
=====================================================================================================================
                                                                      coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------------------------------------------
const                                                              -3.6332      0.034   -107.420      0.000      -3.700      -3.567
lnoperatingmargin                                                   0.2969      0.016     18.678      0.000       0.266       0.328
Agriculture Forestry And Fishing                                    0.6587      0.357      1.847      0.065      -0.040       1.358
Construction                                                        0.3404      0.064      5.292      0.000       0.214       0.467
Finance Insurance And Real Estate                                  -0.1784      0.127     -1.405      0.160      -0.427       0.070
Manufacturing                                                      -0.0182      0.052     -0.351      0.726      -0.120       0.083
Mining                                                             -0.0450      0.048     -0.947      0.344      -0.138       0.048
Retail Trade                                                        0.1256      0.074      1.699      0.089      -0.019       0.270
Services                                                           -0.2630      0.085     -3.105      0.002      -0.429      -0.097
Transportation Communications Electric Gas And Sanitary Service    -0.1285      0.058     -2.215      0.027      -0.242      -0.015
==============================================================================
Omnibus:                    18757.262   Durbin-Watson:                   1.031
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           159948.347
Skew:                           0.533   Prob(JB):                         0.00
Kurtosis:                       8.657   Cond. No.                         51.5
==============================================================================

Warnings:
[1] Standard Errors are robust to cluster correlation (cluster)
```

Estimated t-statistic is several times lower after clustering!

Interaction Variables and Squared Effects

Interaction variables

- The effect of an independent variable may depend on the value of another independent variable in the system
- For instance, a variable on profitability may have a different predictive effect on stock price depending on whether the company’s revenue is growing or not
- Interaction variables allow us to model different slope coefficients in linear systems for different situations
- It is similar to dummy variables, where you can model different intercepts for various situations

```
99 #####INTERACTION VARIABLES###
100 dummies = industrydummies.columns.values
101 dummies = dummies[1:len(dummies)-1]
102 dlist = []
103 for ind in dummies:
104     tag = 'lnoperatingmargin'+'_'+ind
105     data[tag] = data['lnoperatingmargin']*(data[ind])
106     dlist.append(tag)
107     dlist.append(ind)
108 dlist.append('lnoperatingmargin')
109 result = sm.OLS(data['lnepratio'], sm.add_constant(data[dlist]), missing='drop').fit(cov_type='cluster', cov_kwds={'groups': data_w_dummies['siccode']})
110 result.summary()
```

Lines 99 to 107: Generate the interaction variables. In this case, we create interaction variables as lnoperatingmargin*industry dummy

Lines 108 to 110: Run OLS estimation with interaction variables

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             lnepratio   R-squared:                       0.053
Model:                           OLS   Adj. R-squared:                  0.052
Method:                Least Squares   F-statistic:                     22.55
Date:               Fri, 16 Oct 2020   Prob (F-statistic):           2.45e-61
Time:                       20:01:30   Log-Likelihood:                -41725.
No. Observations:              31998   AIC:                         8.348e+04
Df Residuals:                  31982   BIC:                         8.361e+04
Df Model:                         15
Covariance Type:             cluster
======================================================================================================================================================
                                                                                  coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------------------------------------------------------------------------------
const                                                                          -3.6108      0.048    -75.842      0.000      -3.704      -3.518
lnoperatingmargin_Construction                                                  0.2473      0.109      2.266      0.023       0.033       0.461
Construction                                                                    0.7646      0.378      2.021      0.043       0.023       1.506
lnoperatingmargin_Finance Insurance And Real Estate                            -0.1840      0.194     -0.947      0.343      -0.565       0.197
Finance Insurance And Real Estate                                              -0.8341      0.417     -2.002      0.045      -1.651      -0.018
lnoperatingmargin_Manufacturing                                                -0.0364      0.066     -0.555      0.579      -0.165       0.092
Manufacturing                                                                  -0.2274      0.106     -2.151      0.031      -0.435      -0.020
lnoperatingmargin_Mining                                                        0.3526      0.084      4.213      0.000       0.189       0.517
Mining                                                                          0.4767      0.143      3.334      0.001       0.196       0.757
lnoperatingmargin_Retail Trade                                                 -0.0927      0.082     -1.132      0.258      -0.253       0.068
Retail Trade                                                                   -0.2019      0.218     -0.926      0.354      -0.629       0.225
lnoperatingmargin_Services                                                      0.1619      0.060      2.720      0.007       0.045       0.279
Services                                                                       -0.0084      0.132     -0.063      0.950      -0.268       0.251
lnoperatingmargin_Transportation Communications Electric Gas And Sanitary Service  0.0424  0.075     0.567      0.571      -0.104       0.189
Transportation Communications Electric Gas And Sanitary Service                -0.0560      0.163     -0.343      0.732      -0.376       0.264
lnoperatingmargin                                                               0.2399      0.021     11.437      0.000       0.199       0.281
==============================================================================
Omnibus:                    8577.755   Durbin-Watson:                   1.016
Prob(Omnibus):                 0.000   Jarque-Bera (JB):            89878.807
Skew:                          0.988   Prob(JB):                         0.00
Kurtosis:                     10.969   Cond. No.                         144.
==============================================================================

Warnings:
[1] Standard Errors are robust to cluster correlation (cluster)
"""
```

What does the coefficient of each interaction variable mean?

Approximating non-linear relationships with OLS estimations: Squared and higher order terms

We may have convex or concave relationships in finance / economics (e.g. utility functions, industrial cost functions …)

A convex or concave relationship between dependent and independent variable can be modelled by adding squared independent variables to the estimation

Higher order terms are also possible if this is motivated by theory

Note that using log independent variables also models non-linear relationships

```
114 ######SQUARED VARIABLES######
115 #We hypothesize that a stock's valuation has a concave relationship to its asset tangiblility - why?
116 data['tangibles_assets_normal'] = data['tangibles']/data['assets']
117 data['tangibles_assets_sq'] = data['tangibles_assets_normal']*data['tangibles_assets_normal']
118 result = sm.OLS(data['epratio'], sm.add_constant(data[['tangibles_assets_normal', 'tangibles_assets_sq']]), missing='drop').fit(cov_type='cluster', cov_kwds={'groups': da
119 print(result.summary())
120 result = sm.OLS(data['epratio'], sm.add_constant(data[['tangibles_assets_normal']]), missing='drop').fit(cov_type='cluster', cov_kwds={'groups': data_w_dummies['siccode']
121 print(result.summary())
```

- We explore in lines 114 to 121 the hypothesis that a stock's earnings may have a concave relationship with its asset tangibility (plant property equipment / total assets)

- Initially, having more asset tangibility could allow the firm greater access to financing, because the assets can be pledged as collateral

- However, these is an optimal point ("maxima"). Beyond that, having too much fixed assets may mean the firm has less cash on its balance sheet, giving it less freedom to pursue growth opportunities or react to emergencies

- Lines 116 to 117: Generate measures of asset tangibility and squared asset tangibility
- Lines 118 and 120: Run OLS versus only asset tangibility, versus against asset tangibility and asset tangibility_sq

```
Warnings:
[1] Standard Errors are robust to cluster correlation (cluster)
                            OLS Regression Results
==============================================================================
Dep. Variable:                epratio   R-squared:                       0.000
Model:                            OLS   Adj. R-squared:                 -0.000
Method:                 Least Squares   F-statistic:                   0.04372
Date:                Fri, 16 Oct 2020   Prob (F-statistic):              0.834
Time:                        21:42:21   Log-Likelihood:            -1.0468e+05
No. Observations:               31998   AIC:                         2.094e+05
Df Residuals:                   31996   BIC:                         2.094e+05
Df Model:                           1
Covariance Type:              cluster
===========================================================================================
                              coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------------
const                       0.0881      0.123      0.715      0.474      -0.153       0.329
tangibles_assets_normal    -0.0258      0.123     -0.209      0.834      -0.267       0.216
==============================================================================
Omnibus:                   151537.582   Durbin-Watson:                   1.998
Prob(Omnibus):                  0.000   Jarque-Bera (JB):   1359216358122.066
Skew:                         178.596   Prob(JB):                         0.00
Kurtosis:                   31930.236   Cond. No.                         7.73
==============================================================================

Warnings:
[1] Standard Errors are robust to cluster correlation (cluster)
```

A univariate regression masks a more non-linear relationship …

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                epratio   R-squared:                       0.000
Model:                            OLS   Adj. R-squared:                 -0.000
Method:                 Least Squares   F-statistic:                     1.743
Date:                Fri, 16 Oct 2020   Prob (F-statistic):              0.175
Time:                        21:42:21   Log-Likelihood:            -1.0468e+05
No. Observations:               31998   AIC:                         2.094e+05
Df Residuals:                   31995   BIC:                         2.094e+05
Df Model:                           2
Covariance Type:              cluster
===========================================================================================
                              coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------------
const                      -0.1497      0.245     -0.612      0.540      -0.629       0.330
tangibles_assets_normal     0.7366      1.058      0.696      0.486      -1.338       2.811
tangibles_assets_sq        -0.5502      0.850     -0.647      0.517      -2.216       1.116
==============================================================================
Omnibus:                   151535.712   Durbin-Watson:                   1.998
Prob(Omnibus):                  0.000   Jarque-Bera (JB):    1359061937976.077
Skew:                         178.589   Prob(JB):                         0.00
Kurtosis:                   31928.422   Cond. No.                         57.2
==============================================================================

Warnings:
[1] Standard Errors are robust to cluster correlation (cluster)
```

Which we can uncover with a squared term!

Multicollinearity

Multicollinearity

- Multicollinearity occurs when the explanatory variables are very highly correlated with each other.
- Perfect multicollinearity
    - Cannot estimate all the coefficients
        - e.g. suppose $x_3 = 2x_2$
        - and the model is $y_t = \beta_1 + \beta_2 x_{2t} + \beta x_{3t} + \beta_4 x_{4t} + u_t$

- Problems if near multicollinearity is present
    - R^2 will be high but the individual coefficients will have high standard errors.
    - The regression becomes very sensitive to small changes in the specification.
    - Thus confidence intervals for the parameters will be very wide, and significance tests might therefore give inappropriate conclusions.

Measuring Multicollinearity

- The easiest way to measure the extent of multicollinearity is simply to look at the matrix of correlations between the individual variables. e.g.

corr	x_2	x_3	x_4
x_2	–	0.2	0.8
x_3	0.2	–	0.3
x_4	0.8	0.3	–

- But another problem: if 3 or more variables are linear
  – e.g. $x_{2t} + x_{3t} = x_{4t}$,
  then this method will not work.

- Note that high correlation between $y$and one of the x's is not muticollinearity.

Solutions to the Problem of Multicollinearity

Some econometricians argue that if the model is otherwise OK, just ignore it.

Other ways to “cure” the problems are

- drop one of the collinear variables
- transform the highly correlated variables into a ratio
- go out and collect more data e.g.
  - a longer run of data
  - switch to a higher frequency

Challenge or opportunity?

- In our example dataset, there are hundreds of variables.
- This presents two challenges:
  - First, collinearity.
  - Second, it is difficult to interpret the result of extremely high dimensional estimations
- However, the number of variables also allows us to test a larger number of possible hypotheses
- Principal component analysis and/or clustering the data may allow us to leverage the strengths of the dataset while avoiding the above two challenges
- We will detail both of these techniques in class 8 on machine learning and classification

3. Discrete dependent variables

OLS is not the only widely used estimation algorithm

- **Probit / logit models**
  - Used for binary dependent variables
- **Multinomial probit / logit**
  - Categorical dependent variables
- **Ordered probit / logit**
  - Discrete dependent variables

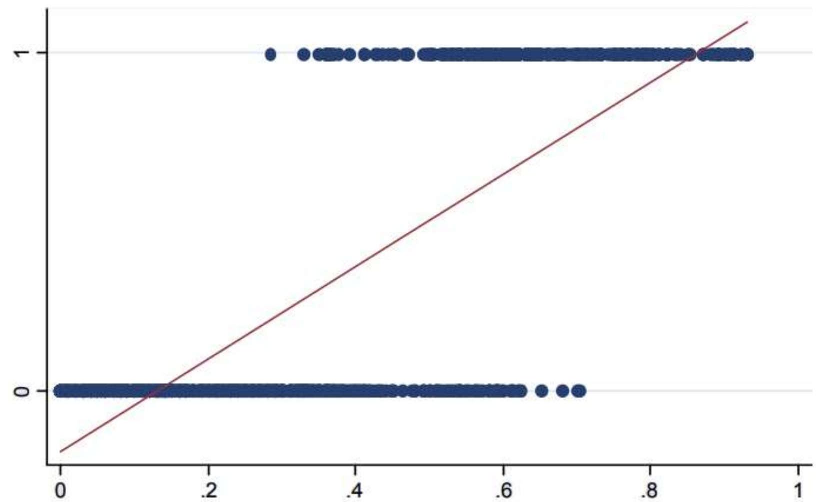
Probit / logit models

$Y_i = \alpha + \beta X_i$ where Y is either 1 or 0

Estimation options:

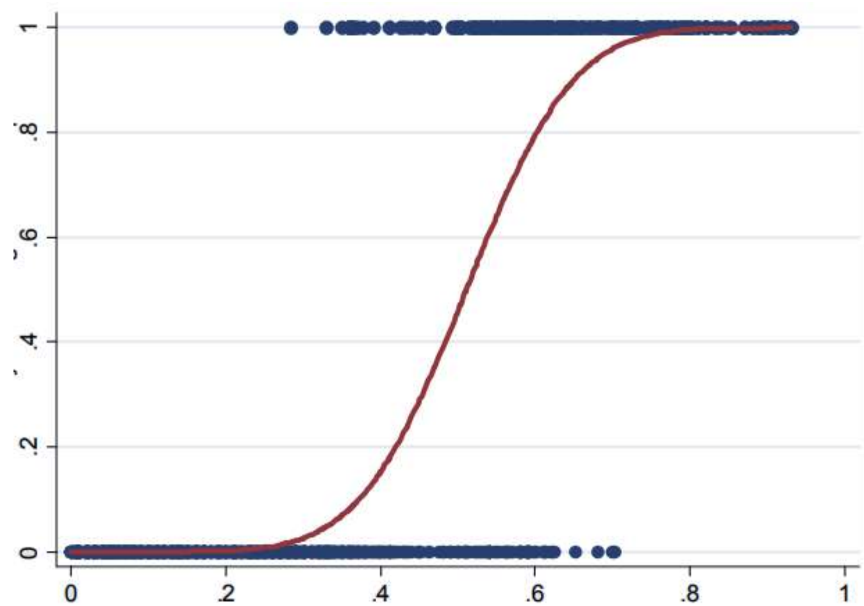
1. Just run OLS; estimate will still be unbiased and BLUE
2. Use a probit or logit model, which exploits fact that Y_i is binary.

Fitting a linear line to a dichotomous variable



Estimated line can continue infinitely above or below [0,1]

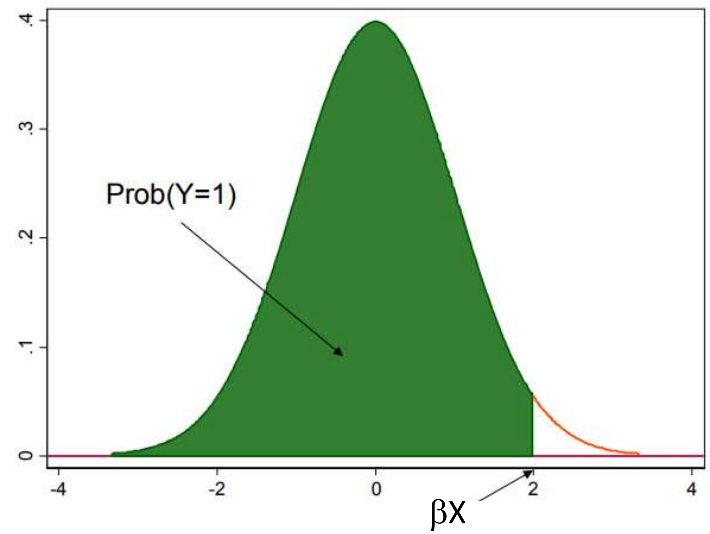
In this case, a nonlinear curve may suit the data better



What curve have we seen in this class which is:

1. Naturally between [0,1]
2. Takes above shape

CDF of normal distribution is used to determine probability of observing a 1 in the probit model



Estimating probit/logit models

Probit models are estimated using maximum likelihood (MLE), which is an iterative computational process

- Tries different values of coefficients repeatedly to estimate the parameters of a probability distribution
- Objective function is to maximize the likelihood of observing the dataset
- For a probit model, a normal distribution is used for the CDF, while in a logit function is used in the case of the logit model
- Density function of the logit model is very similar to standard normal (it may be computationally more efficient), but with thinner tails

Interpreting output of a logit model

01

Estimated coefficients for a logit model are log(odds-ratio)

02

Odds-ratio of an event is p(1-p) where p is the probability of the event

03

For the logit model, p is the probability of observing a 1

Extending concept of discrete dependent variables

- Returning to example of bond ratings, recall we can have 21 bond ratings e.g. with Moodys
- We can code each bond rating as “1”, “2”, “3’, etc in order of their probability of default. E.g. “1” corresponds to “AAA”, “21” to “CCC” etc
- Note there will be a natural ordering, where “1” is “better” than “2” in terms of default probability, etc

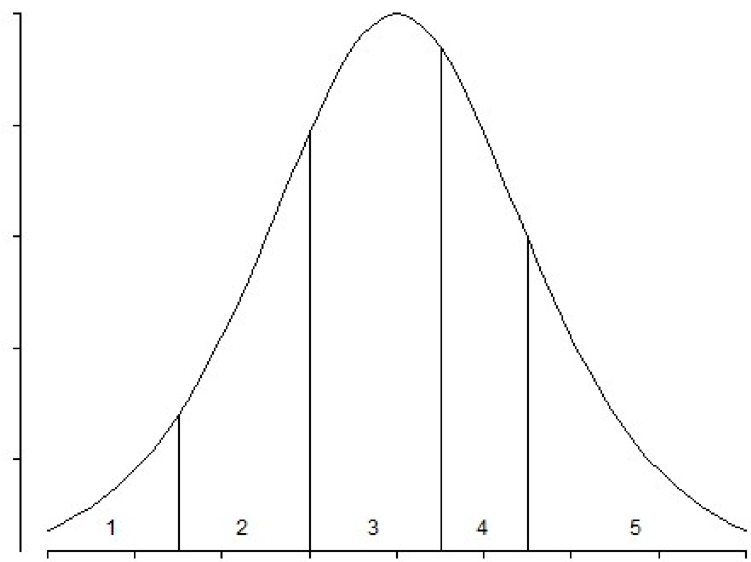
Ordered logit/probit models are a more general case of logit/probit

- Both ordered logit / probit models are a form of classification algorithm where the dependent value exists on an arbitrary scale, where only the relative ordering between different values is significant

- E.g. to model “customer satisfaction” from “5 – very good, 4 – good, 3 – neutral, etc”

- Estimation similar to logit/probit models, except multiple breakpoints on x-axis are estimated instead of just 1

Estimation of ordered logit/probit model by MLE



What if dependent variable is purely categorical?

- It is also possible for dependent variable to be purely discrete and categorical

- In this case, each value of dependent variable is merely a label, and larger or smaller values of dependent variable have no meaning

- For example, we may encode a discrete dependent variable to take values of 1, 2, 3, 4, 5, where 1 = blue, 2 = red, 3 = green, 4 = yellow and 5 = black

- In this case, 1 <2 < 3 < … does not mean anything. The numbers are just **class labels**

We may use multinomial logit and probit models for classification problems

We wish to forecast which class / category a dependent variable will take, given independent variable values

Econometric methodology is to construct a score function that constructs a score from a set of weights that are linearly combined with the explanatory variables (features)

$$\mathrm{score}(\mathbf{X}_i, k) = \boldsymbol{\beta}_k \cdot \mathbf{X}_i$$

Multinomial estimation pseudo-algorithm

- Estimate coefficients for each score function independently (number of score functions is # categories - 1)
- Each score function gives (probability of an outcome) / (probability of base case)
- All probabilities must sum to 1, which gives us probability of base case
- Pick alternative with highest probability
- Estimation framework is also MLE

Methodology roadmap

Nature of Dependent Variable	Methodology
Continuous	OLS Collinearity: Ridge Outliers: Ridge, Lasso or ElasticNet Uni regression & Outliers: Theil-Sen
Discrete 1/0	Probit/Logit
Multivalued Discrete, with a value order [i.e. lower numbers are 'better' or worse']	Ordered Probit/Logit
Categorical labels	Multinomial Probit/Logit