

# Introduction and Python Refresher

## Empirical Industrial Organization

Andras Niedermayer<sup>1</sup>

<sup>1</sup>Department of Economics (THEMA), Cergy Paris Université

Winter 2022



- additional material will be posted at `http://andras.niedermayer.ch/teaching/`
- if you have any questions, write to: `andras.niedermayer@u-cergy.fr`
- please bring along your laptops for the hands-on computer exercises
- we will have a combination of lectures, hands-on exercises in class and take home work
- the grade will be based on a take home exam/term paper

# Examples of Application of Empirical Industrial Organization

- car industry, environmental policy
- auctions
- price discrimination

- merger control

- for example, in 2017 the PSA Group acquired Opel and Vauxhall
- should competition authorities have cleared the acquisition?
- counterfactual: what is the prediction on price changes for the acquisition?

# Examples: Discrete Choice

- merger control
  - for example, in 2017 the PSA Group acquired Opel and Vauxhall
  - should competition authorities have cleared the acquisition?
  - counterfactual: what is the prediction on price changes for the acquisition?

- merger control
  - for example, in 2017 the PSA Group acquired Opel and Vauxhall
  - should competition authorities have cleared the acquisition?
  - counterfactual: what is the prediction on price changes for the acquisition?

- merger control
  - for example, in 2017 the PSA Group acquired Opel and Vauxhall
  - should competition authorities have cleared the acquisition?
  - counterfactual: what is the prediction on price changes for the acquisition?

- **environmental policy**

- for example, France introduced a feebate policy for cars in 2008
- high CO<sub>2</sub> emission cars get taxed, low CO<sub>2</sub> emission cars get a rebate
- the intention was to have a balanced budget



# Examples: Discrete Choice

- environmental policy
  - for example, France introduced a feebate policy for cars in 2008
  - high CO2 emission cars get taxed, low CO2 emission cars get a rebate
  - the intention was to have a balanced budget

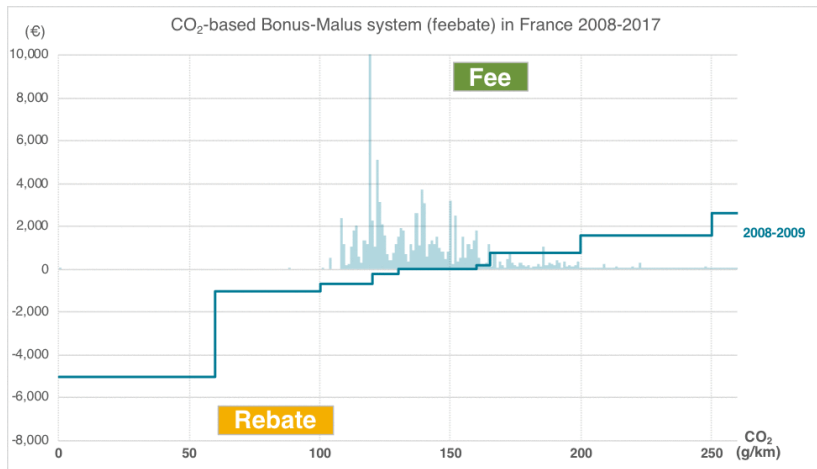
# Examples: Discrete Choice

- environmental policy
  - for example, France introduced a feebate policy for cars in 2008
  - high CO2 emission cars get taxed, low CO2 emission cars get a rebate
  - the intention was to have a balanced budget

# Examples: Discrete Choice

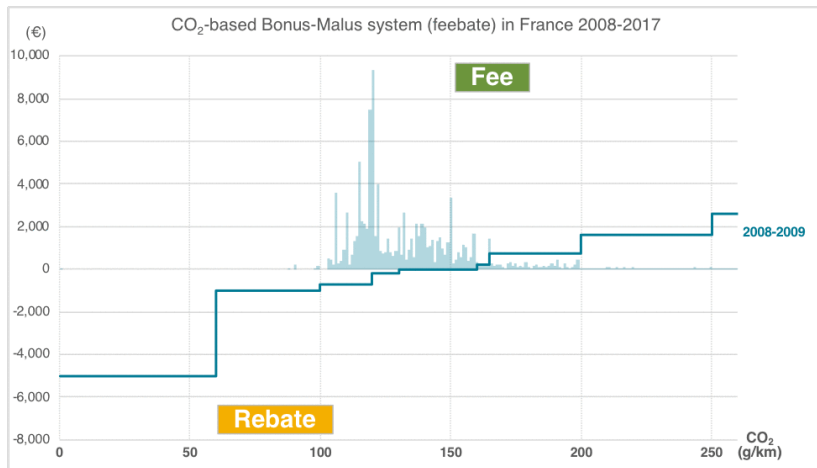
- environmental policy
  - for example, France introduced a feebate policy for cars in 2008
  - high CO2 emission cars get taxed, low CO2 emission cars get a rebate
  - the intention was to have a balanced budget

# Examples: Discrete Choice



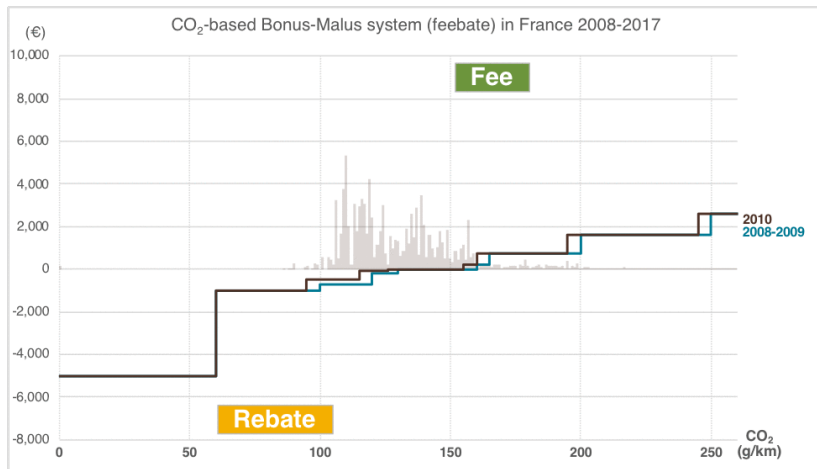
Source: International Council on Clean Transportation

# Examples: Discrete Choice



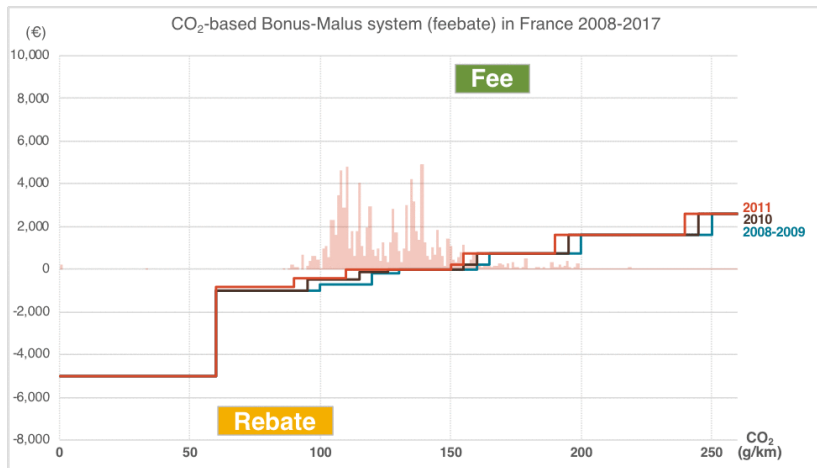
Source: International Council on Clean Transportation

# Examples: Discrete Choice



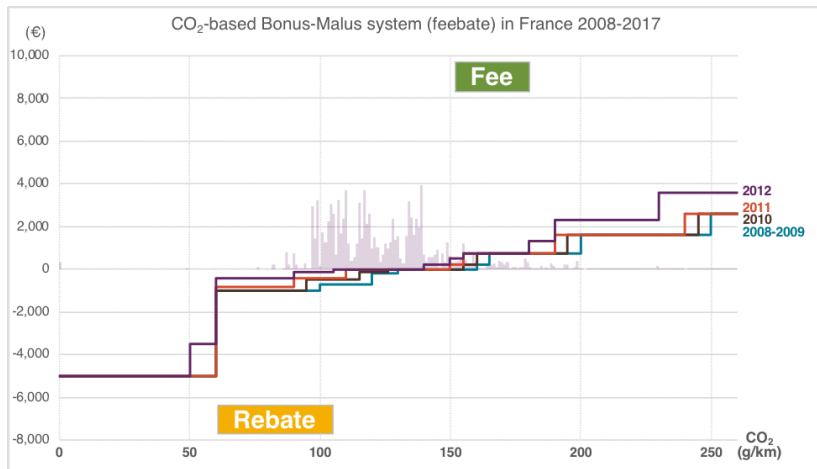
Source: International Council on Clean Transportation

# Examples: Discrete Choice



Source: International Council on Clean Transportation

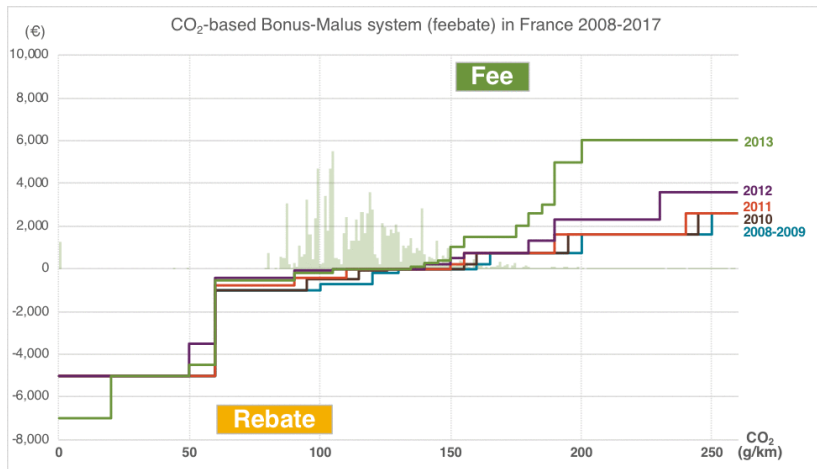
# Examples: Discrete Choice



Source: International Council on Clean Transportation

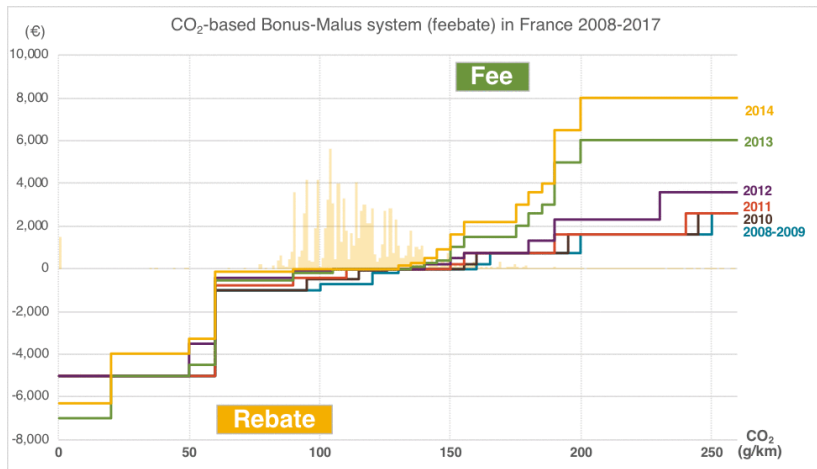


# Examples: Discrete Choice



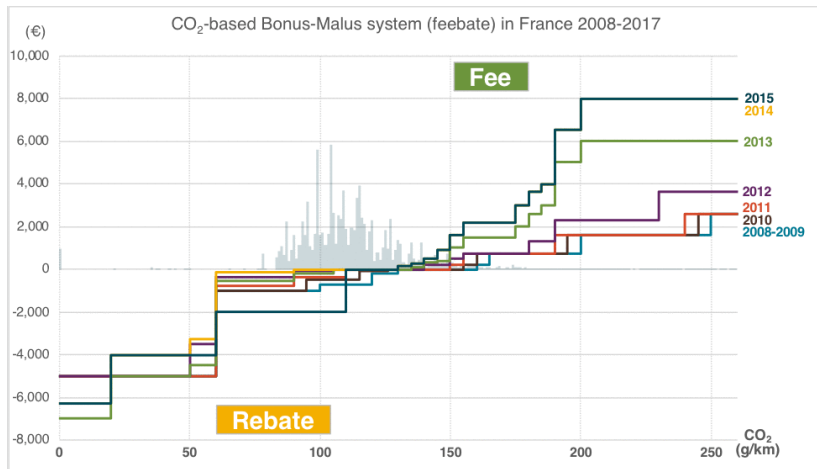
Source: International Council on Clean Transportation

# Examples: Discrete Choice



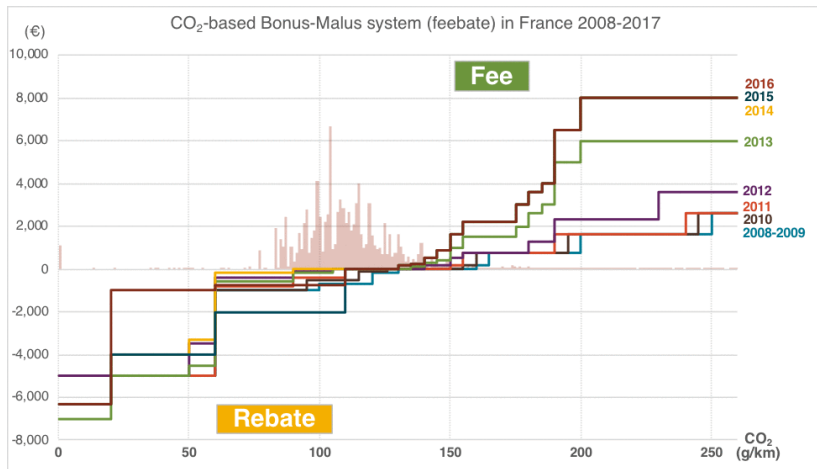
Source: International Council on Clean Transportation

# Examples: Discrete Choice



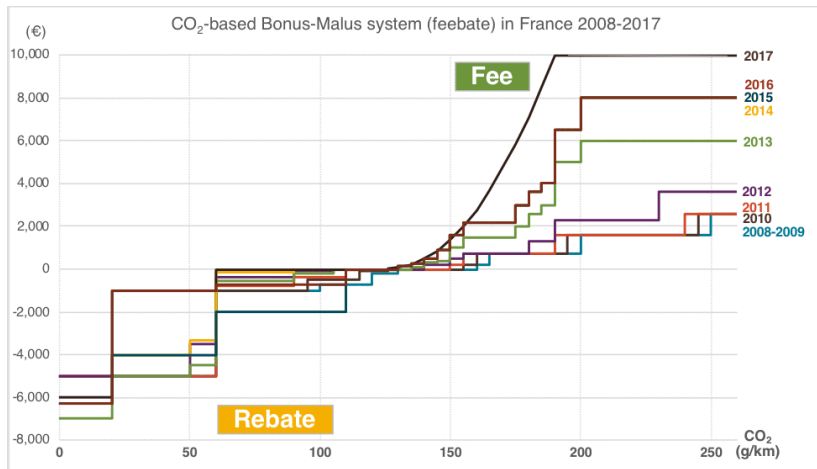
Source: International Council on Clean Transportation

# Examples: Discrete Choice



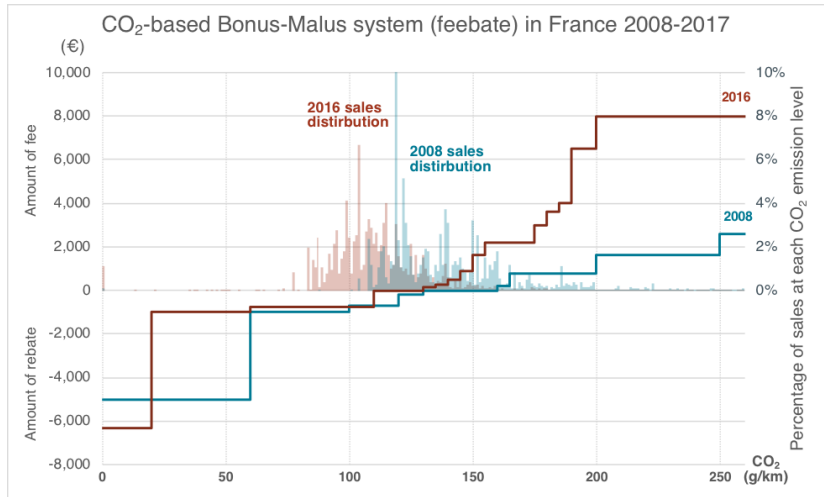
Source: International Council on Clean Transportation

# Examples: Discrete Choice

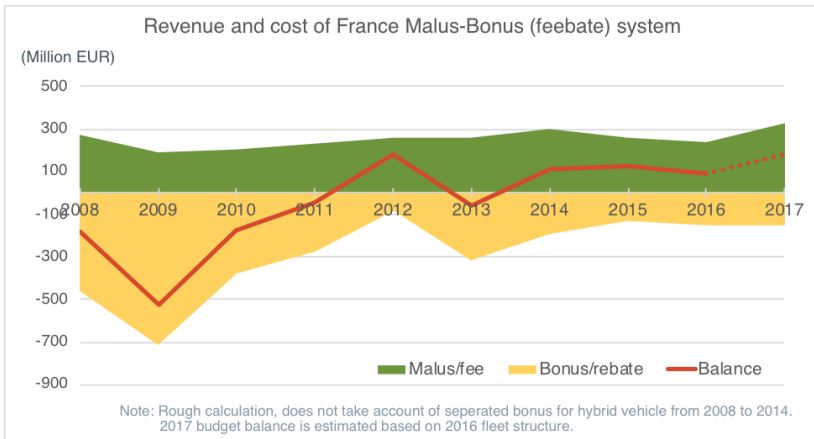


Source: International Council on Clean Transportation

# 2008 vs 2016



Source: International Council on Clean Transportation



Source: International Council on Clean Transportation

# Examples: Auctions





- auctions

- for example, every year that Canadian government auctions off rights to log on government land
- What is the optimal auction format?
- Which minimal price should the government set?

- procurement auctions

- (the government) buys from the lowest bidder on a project, e.g. the construction of roads
- “Operation Hammer” in Quebec, started in 2009: uncovered widespread collusion in the bidding for government construction contracts
- How do you detect collusion?
- How do you compute damages from collusion?

# Examples: Auctions

- auctions

- for example, every year that Canadian government auctions off rights to log on government land
- What is the optimal auction format?
- Which minimal price should the government set?

- procurement auctions

- (the government) buys from the lowest bidder on a project, e.g. the construction of roads
- “Operation Hammer” in Quebec, started in 2009: uncovered widespread collusion in the bidding for government construction contracts
- How do you detect collusion?
- How do you compute damages from collusion?

# Examples: Auctions

- auctions

- for example, every year that Canadian government auctions off rights to log on government land
- What is the optimal auction format?
- Which minimal price should the government set?

- procurement auctions

- (the government) buys from the lowest bidder on a project, e.g. the construction of roads
- “Operation Hammer” in Quebec, started in 2009: uncovered widespread collusion in the bidding for government construction contracts
- How do you detect collusion?
- How do you compute damages from collusion?

# Examples: Auctions

- auctions

- for example, every year that Canadian government auctions off rights to log on government land
- What is the optimal auction format?
- Which minimal price should the government set?

- procurement auctions

- (the government) buys from the lowest bidder on a project, e.g. the construction of roads
- “Operation Hammer” in Quebec, started in 2009: uncovered widespread collusion in the bidding for government construction contracts
- How do you detect collusion?
- How do you compute damages from collusion?

# Examples: Auctions

- auctions
  - for example, every year that Canadian government auctions off rights to log on government land
  - What is the optimal auction format?
  - Which minimal price should the government set?
- procurement auctions
  - (the government) buys from the lowest bidder on a project, e.g. the construction of roads
  - “Operation Hammer” in Quebec, started in 2009: uncovered widespread collusion in the bidding for government construction contracts
  - How do you detect collusion?
  - How do you compute damages from collusion?

# Examples: Auctions

- auctions
  - for example, every year that Canadian government auctions off rights to log on government land
  - What is the optimal auction format?
  - Which minimal price should the government set?
- procurement auctions
  - (the government) buys from the lowest bidder on a project, e.g. the construction of roads
  - “Operation Hammer” in Quebec, started in 2009: uncovered widespread collusion in the bidding for government construction contracts
  - How do you detect collusion?
  - How do you compute damages from collusion?

# Examples: Auctions

- auctions
  - for example, every year that Canadian government auctions off rights to log on government land
  - What is the optimal auction format?
  - Which minimal price should the government set?
- procurement auctions
  - (the government) buys from the lowest bidder on a project, e.g. the construction of roads
  - “Operation Hammer” in Quebec, started in 2009: uncovered widespread collusion in the bidding for government construction contracts
  - How do you detect collusion?
  - How do you compute damages from collusion?

# Examples: Auctions

- auctions
  - for example, every year that Canadian government auctions off rights to log on government land
  - What is the optimal auction format?
  - Which minimal price should the government set?
- procurement auctions
  - (the government) buys from the lowest bidder on a project, e.g. the construction of roads
  - “Operation Hammer” in Quebec, started in 2009: uncovered widespread collusion in the bidding for government construction contracts
  - How do you detect collusion?
  - How do you compute damages from collusion?



# Examples: Auctions

- auctions
  - for example, every year that Canadian government auctions off rights to log on government land
  - What is the optimal auction format?
  - Which minimal price should the government set?
- procurement auctions
  - (the government) buys from the lowest bidder on a project, e.g. the construction of roads
  - “Operation Hammer” in Quebec, started in 2009: uncovered widespread collusion in the bidding for government construction contracts
  - How do you detect collusion?
  - How do you compute damages from collusion?

# Examples: Price Discrimination

## Automobiles



Renault Clio  
€16,600, power: 58 kW

# Examples: Price Discrimination

## Automobiles



€11,300  
power: 43 kW



€16,600  
power: 58 kW



€39,700  
power: 187 kW

# Examples: Price Discrimination

## Automobiles

											
€11,300	€11,875	€12,375	€12,800	€12,833	€12,880	€13,025	€13,172	€13,300	€13,325	€13,433	€13,433
											
€13,625	€13,798	€13,825	€13,875	€13,948	€13,988	€14,200	€14,300	€14,325	€14,333	€14,357	€14,357
											
€14,358	€14,460	€14,466	€14,488	€14,525	€14,600	€14,640	€14,800	€14,820	€15,068	€15,100	€15,100
											
€15,200	€15,298	€15,320	€15,425	€15,448	€15,450	€15,488	€15,625	€15,650	€15,680	€15,700	€15,700
											
€15,713	€15,787	€15,858	€15,960	€15,988	€16,050	€16,070	€16,125	€16,140	€16,180	€16,400	€16,400
											
€16,464	€16,550	€16,600	€16,620	€16,688	€16,733	€16,788	€16,900	€16,964	€17,013	€17,080	€17,080
											
€17,100	€17,120	€17,138	€17,188	€17,212	€17,250	€17,400	€17,413	€17,430	€17,466	€17,513	€17,513
											
€17,560	€17,580	€17,590	€17,712	€17,750	€17,788	€17,820	€17,888	€17,913	€17,930	€18,060	€18,060
											
€18,200	€18,288	€18,320	€18,388	€18,488	€18,675	€18,760	€18,988	€19,175	€19,260	€19,488	€19,488
											
€19,520	€19,679	€19,688	€20,020	€20,039	€20,188	€20,617	€21,200	€21,700	€23,200	€23,750	€23,750
											
€24,367	€36,913	€38,568	€39,700								

# Outlook for Classes

- Python refresher (introduction?)
- discrete choice estimation
- auction econometrics
- econometrics of price discrimination

# Outlook for Classes

- Python refresher (introduction?)
- discrete choice estimation
- auction econometrics
- econometrics of price discrimination

# Outlook for Classes

- Python refresher (introduction?)
- discrete choice estimation
- auction econometrics
- econometrics of price discrimination

# Outlook for Classes

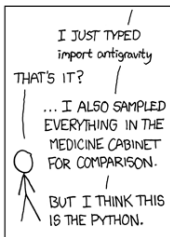
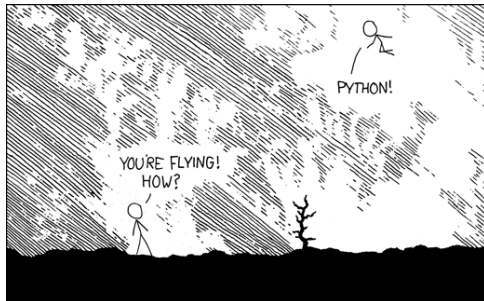
- Python refresher (introduction?)
- discrete choice estimation
- auction econometrics
- econometrics of price discrimination



- Python is the most popular language for (big) data analysis both in academia and industry, see e.g.  
<http://pypl.github.io/PYPL.html>
- interactive data analytics: immediate feedback and intuitive data processing in Jupyter notebooks (and also IPython console and interpreters in various Integrated Development environments)

- Python is the most popular language for (big) data analysis both in academia and industry, see e.g.  
`http://pypl.github.io/PYPL.html`
- interactive data analytics: immediate feedback and intuitive data processing in Jupyter notebooks (and also IPython console and interpreters in various Integrated Development environments)

- powerful libraries



- Open source: free, powerful, and a very dynamic community to develop new packages
- Cross-platform: for Windows, OS X, and Linux
- Intuitive language: Closest to mathematical language and pseudocode
- Dynamically typed: One does not need to declare variable types statically (in new versions of Python one can optionally declare the types of variables, which is an advantage in large code bases, but this is outside of the scope of this lecture)

- Open source: free, powerful, and a very dynamic community to develop new packages
- Cross-platform: for Windows, OS X, and Linux
- Intuitive language: Closest to mathematical language and pseudocode
- Dynamically typed: One does not need to declare variable types statically (in new versions of Python one can optionally declare the types of variables, which is an advantage in large code bases, but this is outside of the scope of this lecture)

- Open source: free, powerful, and a very dynamic community to develop new packages
- Cross-platform: for Windows, OS X, and Linux
- Intuitive language: Closest to mathematical language and pseudocode
- Dynamically typed: One does not need to declare variable types statically (in new versions of Python one can optionally declare the types of variables, which is an advantage in large code bases, but this is outside of the scope of this lecture)

- Open source: free, powerful, and a very dynamic community to develop new packages
- Cross-platform: for Windows, OS X, and Linux
- Intuitive language: Closest to mathematical language and pseudocode
- Dynamically typed: One does not need to declare variable types statically (in new versions of Python one can optionally declare the types of variables, which is an advantage in large code bases, but this is outside of the scope of this lecture)

- for initial lectures: run code online in Jupyter notebooks on Sargent and Stachurski's website: [https://python.quantecon.org/index\\_postgrad.html](https://python.quantecon.org/index_postgrad.html)
  - click on "Launch notebook" to run a notebook
  - run the code
  - you can edit the code and rerun it
  - to save the code, you need a Google account
- you can create you own online Jupyter notebooks on Google Colaboratory  
<https://colab.research.google.com/>
- for later lecture, it's easier to run you code locally, so please install Anaconda Python 3.9 on your laptops,  
<https://www.anaconda.com/distribution/>



- for initial lectures: run code online in Jupyter notebooks on Sargent and Stachurski's website: [https://python.quantecon.org/index\\_postgrad.html](https://python.quantecon.org/index_postgrad.html)
  - click on "Launch notebook" to run a notebook
    - run the code
    - you can edit the code and rerun it
    - to save the code, you need a Google account
- you can create you own online Jupyter notebooks on Google Colaboratory  
<https://colab.research.google.com/>
- for later lecture, it's easier to run you code locally, so please install Anaconda Python 3.9 on your laptops,  
<https://www.anaconda.com/distribution/>

- for initial lectures: run code online in Jupyter notebooks on Sargent and Stachurski's website: [https://python.quantecon.org/index\\_postgrad.html](https://python.quantecon.org/index_postgrad.html)
  - click on "Launch notebook" to run a notebook
  - run the code
    - you can edit the code and rerun it
    - to save the code, you need a Google account
- you can create you own online Jupyter notebooks on Google Colaboratory  
<https://colab.research.google.com/>
- for later lecture, it's easier to run you code locally, so please install Anaconda Python 3.9 on your laptops,  
<https://www.anaconda.com/distribution/>

- for initial lectures: run code online in Jupyter notebooks on Sargent and Stachurski's website: [https://python.quantecon.org/index\\_postgrad.html](https://python.quantecon.org/index_postgrad.html)
  - click on "Launch notebook" to run a notebook
  - run the code
  - you can edit the code and rerun it
  - to save the code, you need a Google account
- you can create you own online Jupyter notebooks on Google Colaboratory  
<https://colab.research.google.com/>
- for later lecture, it's easier to run you code locally, so please install Anaconda Python 3.9 on your laptops,  
<https://www.anaconda.com/distribution/>

- for initial lectures: run code online in Jupyter notebooks on Sargent and Stachurski's website: [https://python.quantecon.org/index\\_postgrad.html](https://python.quantecon.org/index_postgrad.html)
  - click on "Launch notebook" to run a notebook
  - run the code
  - you can edit the code and rerun it
  - to save the code, you need a Google account
- you can create you own online Jupyter notebooks on Google Colaboratory  
<https://colab.research.google.com/>
- for later lecture, it's easier to run you code locally, so please install Anaconda Python 3.9 on your laptops,  
<https://www.anaconda.com/distribution/>

- for initial lectures: run code online in Jupyter notebooks on Sargent and Stachurski's website: [https://python.quantecon.org/index\\_postgrad.html](https://python.quantecon.org/index_postgrad.html)
  - click on "Launch notebook" to run a notebook
  - run the code
  - you can edit the code and rerun it
  - to save the code, you need a Google account
- you can create you own online Jupyter notebooks on Google Colaboratory  
<https://colab.research.google.com/>
- for later lecture, it's easier to run you code locally, so please install Anaconda Python 3.9 on your laptops,  
<https://www.anaconda.com/distribution/>

- for initial lectures: run code online in Jupyter notebooks on Sargent and Stachurski's website: `https://python.quantecon.org/index_postgrad.html`
  - click on "Launch notebook" to run a notebook
  - run the code
  - you can edit the code and rerun it
  - to save the code, you need a Google account
- you can create you own online Jupyter notebooks on Google Colaboratory  
`https://colab.research.google.com/`
- for later lecture, it's easier to run you code locally, so please install Anaconda Python 3.9 on your laptops,  
`https://www.anaconda.com/distribution/`

- we will start with Python basics
- work through the notebooks on `https://python.quantecon.org/index\_learning\_python.html`
- solve the exercises at the end of the notebooks

- we will start with Python basics
- work through the notebooks on [https://python.quantecon.org/index\\_learning\\_python.html](https://python.quantecon.org/index_learning_python.html)
- solve the exercises at the end of the notebooks



- we will start with Python basics
- work through the notebooks on `https://python.quantecon.org/index\_learning\_python.html`
- solve the exercises at the end of the notebooks

- next we will look at the scientific libraries for Python
- work through the notebooks on  
`https://python.quantecon.org/index\_python\_scientific\_libraries.html`
- solve the exercises at the end of the notebooks

- next we will look at the scientific libraries for Python
- work through the notebooks on  
`https://python.quantecon.org/index\_python\_scientific\_libraries.html`
- solve the exercises at the end of the notebooks

- next we will look at the scientific libraries for Python
- work through the notebooks on  
`https://python.quantecon.org/index\_python\_scientific\_libraries.html`
- solve the exercises at the end of the notebooks