



# SPHINX

Github

 Github

## PYTHON

Github

 Github

## DOCUMENTATION

## GENERATOR

**Sphinx with Github Webpages**

*Release 1.00*

**Wenqiang Feng**

March 17, 2019



# CONTENTS

<b>1 Preface</b>	<b>3</b>
1.1 About . . . . .	3
1.2 Motivation for this tutorial . . . . .	4
1.3 Feedback and suggestions . . . . .	5
<b>2 Introduction</b>	<b>7</b>
2.1 Sphinx: Python Documentation Generator . . . . .	7
2.2 reStructured Text . . . . .	8
2.3 Latex Document Preparation System . . . . .	8
<b>3 Packages Installation</b>	<b>9</b>
3.1 Python Installation . . . . .	9
3.2 Sphinx Installation . . . . .	9
3.3 Latex Installation . . . . .	9
<b>4 Sphinx Configuration</b>	<b>11</b>
4.1 General HTML Configuration . . . . .	11
4.2 General LaTex Configuration . . . . .	14
4.3 Full <code>conf.py</code> Script . . . . .	15
4.4 General Documentation Generator Configuration . . . . .	23
4.5 Full <code>docgen.py</code> Script . . . . .	25
<b>5 reStructuredText Markup</b>	<b>29</b>
5.1 reStructuredText Primer . . . . .	29
5.2 Roles . . . . .	50
5.3 Directives . . . . .	52
<b>6 API Example</b>	<b>55</b>
6.1 <code>rnorm</code> . . . . .	55
6.2 <code>dnorm</code> . . . . .	55
6.3 <code>runif</code> . . . . .	56
6.4 T-test . . . . .	56
<b>7 Publishing on Github</b>	<b>57</b>
7.1 Create reStructuredTexts . . . . .	57
7.2 Create Repository on Github . . . . .	58

7.3	Commit reStructuredTexts folder to Repository . . . . .	58
7.4	Setup Github Pages on Github . . . . .	59
<b>8</b>	<b>Main Reference</b>	<b>63</b>
<b>9</b>	<b>Appendix: My LaTex Cheat Sheet</b>	<b>65</b>
	<b>Bibliography</b>	<b>67</b>
	<b>Python Module Index</b>	<b>69</b>
	<b>Index</b>	<b>71</b>

GitHub



GitHub



# SPHINX

GitHub



## PYTHON DOCUMENTATION GENERATOR

GitHub



Welcome to my **Sphinx github webpage** tutorials! In those tutorials, you will learn how to use Sphinx to create .html and .pdf and how to hookup your Sphinx webpage to github. The PDF version can be downloaded from [HERE](#).



---

# CHAPTER ONE

---

## PREFACE

---

### Chinese proverb

Good tools are prerequisite to the successful execution of a job. – old Chinese proverb

---

## 1.1 About

### 1.1.1 About this tutorial

This document is a summary of my valueable experiences in using Python documentation Sphinx with Github webpage. The PDF version can be downloaded from [HERE](#). **You may download and distribute it. Please be aware, however, that the note contains typos as well as inaccurate or incorrect description.**

In this repository, I try to use the detailed demo code and examples to show how to use Sphinx to generate the .html and .pdf documents and how to hookup them automatically on Github. If you find your work wasn't cited in this note, please feel free to let me know.

Although I am by no means a python programming and Sphinx expert, I decided that it would be useful for me to share what I learned about Sphinx in the form of easy tutorials with detailed example. I hope those tutorials will be a valuable tool for your studies.

The tutorials assume that the reader has a preliminary knowledge of python programing, LaTex and Linux. And this document is generated automatically by using [sphinx](#).

### 1.1.2 About the authors

- **Wenqiang Feng**
  - Data Scientist and PhD in Mathematics
  - University of Tennessee at Knoxville
  - Webpage: <http://web.utk.edu/~wfeng1/>
  - Email: [von198@gmail.com](mailto:von198@gmail.com)

- **Biography**

Wenqiang Feng is Data Scientist within DST's Applied Analytics Group. Dr. Feng's responsibilities include providing DST clients with access to cutting-edge skills and technologies, including Big Data analytic solutions, advanced analytic and data enhancement techniques and modeling.

Dr. Feng has deep analytic expertise in data mining, analytic systems, machine learning algorithms, business intelligence, and applying Big Data tools to strategically solve industry problems in a cross-functional business. Before joining DST, Dr. Feng was an IMA Data Science Fellow at The Institute for Mathematics and its Applications (IMA) at the University of Minnesota. While there, he helped startup companies make marketing decisions based on deep predictive analytics.

Dr. Feng graduated from University of Tennessee, Knoxville, with Ph.D. in Computational Mathematics and Master's degree in Statistics. He also holds Master's degree in Computational Mathematics from Missouri University of Science and Technology (MST) and Master's degree in Applied Mathematics from the University of Science and Technology of China (USTC).

- **Declaration**

The work of Wenqiang Feng was supported by the IMA, while working at IMA. However, any opinion, finding, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the IMA, UTK and DST.

## 1.2 Motivation for this tutorial

`Sphinx` is an awesome Python documentation package, and it has excellent facilities for the documentation of software projects in a range of languages. I have been using `Sphinx` for almost 4 years. I was impressed and attracted by `Sphinx` in the first using. And I foud that:

- It supports **several popular output formats**: HTML (including Windows HTML Help), `LaTeX` (for printable PDF versions), `ePub`, `Texinfo`, manual pages, plain text.
- It has **easy publishing routes**: Github.
- It has **extensive cross-references**: semantic markup and automatic links for functions, classes, citations, glossary terms and similar pieces of information
- It has **hierarchical structure**: easy definition of a document tree, with automatic links to siblings, parents and children.
- It has **automatic indices**: general index as well as a language-specific module indices
- It has awesome **code handling**: automatic highlighting using the Pygments highlighter
- It has abundant **extensions**: automatic testing of code snippets, inclusion of docstrings from Python modules (API docs), and more
- It has abundant **contributed extensions**: more than 50 extensions contributed by users in a second repository; most of them installable from PyPI

## 1.3 Feedback and suggestions

Your comments and suggestions are highly appreciated. I am more than happy to receive corrections, suggestions or feedbacks through email (Wenqiang Feng: [von198@gmail.com](mailto:von198@gmail.com)) for improvements.



---

## CHAPTER TWO

---

## INTRODUCTION

---

### Chinese proverb

If you only know yourself, but not your opponent, you may win or may lose. If you know neither yourself nor your enemy, you will always endanger yourself. – idiom, from Sunzi's Art of War

---

## 2.1 Sphinx: Python Documentation Generator

The following descriptions are from [Sphinx](#):

Sphinx is a tool that makes it easy to create intelligent and beautiful documentation, written by Georg Brandl and licensed under the BSD license.

It was originally created for the Python documentation, and it has excellent facilities for the documentation of software projects in a range of languages. Of course, this site is also created from reStructuredText sources using Sphinx! The following features should be highlighted:

- **Output formats:** HTML (including Windows HTML Help), LaTeX (for printable PDF versions), ePub, Texinfo, manual pages, plain text
- **Extensive cross-references:** semantic markup and automatic links for functions, classes, citations, glossary terms and similar pieces of information
- **Hierarchical structure:** easy definition of a document tree, with automatic links to siblings, parents and children
- **Automatic indices:** general index as well as a language-specific module indices
- **Code handling:** automatic highlighting using the Pygments highlighter
- **Extensions:** automatic testing of code snippets, inclusion of docstrings from Python modules (API docs), and more
- **Contributed extensions:** more than 50 extensions contributed by users in a second repository; most of them installable from PyPI

Sphinx uses [reStructuredText](#) as its markup language, and many of its strengths come from the power and straightforwardness of reStructuredText and its parsing and translating suite, the Docutils.

## 2.2 reStructured Text

The following descriptions are from [reStructuredText](#):

[reStructuredText](#) (RST, ReST, or reST) is a file format for textual data used primarily in the Python programming language community for technical documentation.

## 2.3 Latex Document Preparation System

The following descriptions are from [LaTeX](#):

[LaTeX](#) (a shortening of Lampert TeX) is a document preparation system. When writing, the writer uses plain text as opposed to the formatted text found in WYSIWYG (“what you see is what you get”) word processors like Microsoft Word, LibreOffice Writer and Apple Pages.

LaTeX is widely used in academia for the communication and publication of scientific documents in many fields, including mathematics, statistics, computer science, engineering, chemistry, physics, economics, linguistics, quantitative psychology, philosophy, and political science.

More information can be get from [LaTeX](#).

---

CHAPTER  
THREE

---

## PACKAGES INSTALLATION

---

### Chinese proverb

A journey of a thousand miles begins with a single step – old Chinese proverb

---

**Warning:** It's been 10 years since I abandoned Windows operating systems. So I am a noob for Windows operating systems and I really do not know how to install some packages on Windows operating systems.

### 3.1 Python Installation

1. Install pip:

```
sudo easy_install pip
```

2. Install python:

```
pip install python
```

### 3.2 Sphinx Installation

```
pip install -U Sphinx
```

### 3.3 Latex Installation

You can download the MacTex from: <https://www.tug.org/mactex/> and install it for Mac system. Or you can use the following command to install TexLive on Linux system:

```
sudo apt update && sudo apt install texlive-full
```

## SPHINX CONFIGURATION

**Warning:** I heavily borrowed, modified and used the configuration in `conf.py` and `docgen.py` of [Theano package project](#). I will keep all the comments from Theano team and the copyrights of those files belong to Theano team.

### 4.1 General HTML Configuration

#### 4.1.1 General Environment Infomation

##### 1. Sphinx extension

```
# Add any Sphinx extension module names here, as strings. They can be
# extensions coming with Sphinx (named 'sphinx.ext.*') or your custom ones.
extensions = ['sphinx.ext.autodoc',
              'sphinx.ext.todo',
              'sphinx.ext.doctest',
              'sphinx.ext.napoleon',
              'sphinx.ext.linkcode']

todo_include_todos = True
napoleon_google_docstring = False
napoleon_include_special_with_doc = False
```

##### 2. Math formula support

```
# We do it like this to support multiple sphinx version without having warning.
# Our buildbot consider warning as error.
try:
    from sphinx.ext import imgmath
    extensions.append('sphinx.ext.imgmath')
except ImportError:
    try:
        from sphinx.ext import pngmath
        extensions.append('sphinx.ext.pngmath')
    except ImportError:
        pass
```

### 3. Included and excluded folders options

```
# Add any paths that contain templates here, relative to this directory.
templates_path = ['.templates']

# List of directories, relative to source directories, that shouldn't be
# searched for source files.
exclude_dirs = ['images', 'scripts', 'sandbox']
```

### 4. Image math formula preamble

```
# If false, no module index is generated.
latex_use_modindex = True

default_role = 'math'
pngmath_divpng_args = ['-gamma 1.5', '-D 110']
#pngmath_divpng_args = ['-gamma', '1.5', '-D', '110', '-bg', 'Transparent']
imgmath_latex_preamble = '\\usepackage{amsmath}\n' + \
    '\\usepackage{mathtools}\n' + \
    '\\usepackage{amsfonts}\n' + \
    '\\usepackage{amssymb}\n' + \
    '\\usepackage{dsfont}\n' + \
    '\\def\\Z{\\mathbb{Z}}\n' + \
    '\\def\\R{\\mathbb{R}}\n' + \
    '\\def\\bX{\\mathbf{X}}\n' + \
    '\\def\\X{\\mathbf{X}}\n' + \
    '\\def\\By{\\mathbf{y}}\n' + \
    '\\def\\Bbeta{\\boldsymbol{\\beta}}\n' + \
    '\\def\\U{\\mathbf{U}}\n' + \
    '\\def\\V{\\mathbf{V}}\n' + \
    '\\def\\V1{\\mathbf{1}}\n' + \
    '\\def\\hU{\\mathbf{\\hat{U}}}\n' + \
    '\\def\\hS{\\mathbf{\\hat{\Sigma}}}\n' + \
    '\\def\\hV{\\mathbf{\\hat{V}}}\n' + \
    '\\def\\E{\\mathbf{E}}\n' + \
    '\\def\\F{\\mathbf{F}}\n' + \
    '\\def\\x{\\mathbf{x}}\n' + \
    '\\def\\h{\\mathbf{h}}\n' + \
    '\\def\\v{\\mathbf{v}}\n' + \
    '\\def\\nv{\\mathbf{v}^{\\top}}\n' + \
    '\\def\\nh{\\mathbf{h}^\\top}\n' + \
    '\\def\\s{\\mathbf{s}}\n' + \
    '\\def\\b{\\mathbf{b}}\n' + \
    '\\def\\c{\\mathbf{c}}\n' + \
    '\\def\\W{\\mathbf{W}}\n' + \
    '\\def\\C{\\mathbf{C}}\n' + \
    '\\def\\P{\\mathbf{P}}\n' + \
    '\\def\\T{\\mathcal{T}}\n' + \
    '\\def\\B{\\mathcal{B}}\n'
```

## 4.1.2 General Project Infomation

### 1. The suffix of source filenames

```
# The suffix of source filenames.
source_suffix = '.rst'
```

### 2. The master toctree document

```
# The master toctree document.
master_doc = 'index'
```

### 3. General substitutions

```
# General substitutions.
project = 'Sphinx with Github Webpages'
copyright = '2019, Wenqiang Feng'
```

### 4. Version and date format

```
# We need this hokey-pokey because versioneer needs the current
# directory to be the root of the project to work.
# The short X.Y version.
version = '1.00'
# The full version, including alpha/beta/rc tags.
release = '1.00'

# There are two options for replacing |today|: either, you set today to some
# non-false value, then it is used:
#today = ''
# Else, today_fmt is used as the format for a strftime call.
today_fmt = '%B %d, %Y'

# Add any paths that contain custom static files (such as style sheets) here,
# relative to this directory. They are copied after the builtin static files,
# so a file named "default.css" will overwrite the builtin "default.css".
html_static_path = ['images']

# If not '', a 'Last updated on:' timestamp is inserted at every page bottom,
# using the given strftime format.
html_last_updated_fmt = '%b %d, %Y'

# If true, SmartyPants will be used to convert quotes and dashes to
# typographically correct entities.
html_use_smartytags = True
```

## 4.2 General LaTex Configuration

### 4.2.1 General LaTeX Output Options

```
# Options for LaTeX output
# -----
#
latex_elements = {
    # The paper size ('letter' or 'a4').
    # latex_paper_size = 'a4',
    #
    # The font size ('10pt', '11pt' or '12pt').
    'pointsize': '12pt',
    #
    # Additional stuff for the LaTeX preamble.
    # latex_preamble = '',
}

# Grouping the document tree into LaTeX files. List of tuples
# (source start file, target name, title, author, document class
# [howto/manual]).
latex_documents = [
    ('index', 'sphinxgithub.tex', 'Sphinx Github Webpage Tutorials',
     'Wenqiang Feng', 'manual'),
]
# The name of an image file (relative to this directory) to place at the top
# of
# the title page.
latex_logo = 'images/logo.png'
```

### 4.2.2 LaTeX preamble definitions

```
# latex_elements['preamble'] = '\usepackage{xcolor}'
# Additional stuff for the LaTeX preamble.
# latex_preamble
latex_elements['preamble'] = '\\usepackage{amsmath}\n'+\
    '\\usepackage{mathtools}\n'+\
    '\\usepackage{amsfonts}\n'+\
    '\\usepackage{amssymb}\n'+\
    '\\usepackage{dsfont}\n'+\
    '\\def\\Z{\\mathbb{Z}}\n'+\
    '\\def\\R{\\mathbb{R}}\n'+\
    '\\def\\bx{\\mathbf{X}}\n'+\
    '\\def\\X{\\mathbf{X}}\n'+\
    '\\def\\By{\\mathbf{y}}\n'+\
    '\\def\\Bbeta{\\boldsymbol{\\beta}}\n'+\
    '\\def\\bU{\\mathbf{U}}\n'+\
    '\\def\\bV{\\mathbf{V}}\n'+\
    '\\def\\V1{\\mathbf{1}}\n'+\
    '\\def\\hU{\\mathbf{\hat{U}}}\n'+\
```

(continues on next page)

(continued from previous page)

```
'\\def\\hS{\\mathbf{\\hat{\\Sigma}}}\\n'+\
'\\def\\hV{\\mathbf{\\hat{V}}}\\n'+\
'\\def\\E{\\mathbf{E}}\\n'+\
'\\def\\F{\\mathbf{F}}\\n'+\
'\\def\\x{\\mathbf{x}}\\n'+\
'\\def\\h{\\mathbf{h}}\\n'+\
'\\def\\v{\\mathbf{v}}\\n'+\
'\\def\\nv{\\mathbf{v}^{\\{-\\}}}\\n'+\
'\\def\\nh{\\mathbf{h}^{\\{-\\}}}\\n'+\
'\\def\\s{\\mathbf{s}}\\n'+\
'\\def\\b{\\mathbf{b}}\\n'+\
'\\def\\c{\\mathbf{c}}\\n'+\
'\\def\\W{\\mathbf{W}}\\n'+\
'\\def\\C{\\mathbf{C}}\\n'+\
'\\def\\P{\\mathbf{P}}\\n'+\
'\\def\\T{\\mathbf{T}}\\n'+\
'\\def\\B{\\mathbf{B}}\\n'
```

## 4.3 Full conf.py Script

```
# -*- coding: utf-8 -*-
#####
# I heavily borrowed, modified and used the configuration in conf.py of Theano
# package project. I will keep all the comments from Theano team and the
# copyright of this file belongs to Theano team.
# reference:
#
# Theano repository: https://github.com/Theano/Theano
# conf.py: https://github.com/Theano/Theano/blob/master/doc/conf.py
#####
# theano documentation build configuration file, created by
# sphinx-quickstart on Tue Oct  7 16:34:06 2008.
#
# This file is execfile()d with the current directory set to its containing
# directory.
#
# The contents of this file are pickled, so don't put values in the namespace
# that aren't pickleable (module imports are okay, they're removed
# automatically).
#
# All configuration values have a default value; values that are commented out
# serve to show the default value.
#
# If your extensions are in another directory, add it here. If the directory
# is relative to the documentation root, use os.path.abspath to make it
# absolute, like shown here.
#sys.path.append(os.path.abspath('some/directory'))

from __future__ import absolute_import, print_function, division
```

(continues on next page)

(continued from previous page)

```
import os
import sys

# theano_path = os.path.join(os.path.dirname(__file__), os.pardir)
# sys.path.append(os.path.abspath(theano_path))
sys.path.insert(0, os.path.abspath('..'))
import versioneer

# General configuration
# -------

# Add any Sphinx extension module names here, as strings. They can be
# extensions coming with Sphinx (named 'sphinx.ext.*') or your custom ones.
extensions = ['sphinx.ext.autodoc',
              'sphinx.ext.todo',
              'sphinx.ext.doctest',
              'sphinx.ext.napoleon',
              'sphinx.ext.linkcode',
              'sphinx.ext.intersphinx'
]

todo_include_todos = True
napoleon_google_docstring = False
napoleon_include_special_with_doc = False

# We do it like this to support multiple sphinx version without having warning.
# Our buildbot consider warning as error.
try:
    from sphinx.ext import imgmath
    extensions.append('sphinx.ext.imgmath')
except ImportError:
    try:
        from sphinx.ext import pngmath
        extensions.append('sphinx.ext.pngmath')
    except ImportError:
        pass

# Add any paths that contain templates here, relative to this directory.
#templates_path = ['.templates']

# The suffix of source filenames.
source_suffix = '.rst'

# The master toctree document.
master_doc = 'index'

# General substitutions.
project = 'Sphinx with Github Webpages'
```

(continues on next page)

(continued from previous page)

```

copyright = '2019, Wenqiang Feng'

# The default replacements for |version| and |release|, also used in various
# other places throughout the built documents.
#
# We need this hokey-pokey because versioneer needs the current
# directory to be the root of the project to work.
# The short X.Y version.
version = '1.00'
# The full version, including alpha/beta/rc tags.
release = '1.00'

# There are two options for replacing |today|: either, you set today to some
# non-false value, then it is used:
#today = ''
# Else, today_fmt is used as the format for a strftime call.
today_fmt = '%B %d, %Y'

# List of documents that shouldn't be included in the build.
#unused_docs = []

# List of directories, relative to source directories, that shouldn't be
# searched for source files.
exclude_dirs = ['images', 'scripts', 'sandbox']

# The reST default role (used for this markup: `text`) to use for all
# documents.
#default_role = None

# If true, '()' will be appended to :func: etc. cross-reference text.
#add_function_parentheses = True

# If true, the current module name will be prepended to all description
# unit titles (such as .. function::).
#add_module_names = True

# If true, sectionauthor and moduleauthor directives will be shown in the
# output. They are ignored by default.
#show_authors = False

# The name of the Pygments (syntax highlighting) style to use.
pygments_style = 'sphinx'

# Options for HTML output
# -------

# Enable link of 'View page source'
#html_show_sourcelink = False
# Add 'Edit on Github' link instead of 'View page source'
# reference:https://docs.readthedocs.io/en/latest/vcs.html

```

(continues on next page)

(continued from previous page)

```
html_context = {
    # Enable the "Edit in GitHub link within the header of each page.
    'display_github': True,
    # Set the following variables to generate the resulting github URL for_
    ↪each page.
    # Format Template: https://{{ github_host|default("github.com") }}/{{_
    ↪github_user }}
    #/{{ github_repo }}/blob/{{ github_version }}{{ conf_py_path }}{{_
    ↪pagename }}{{ suffix }}}
    #https://github.com/runawayhorse001/SphinxGithub/blob/master/doc/index.rst
    'github_user': 'runawayhorse001',
    'github_repo': 'SphinxGithub',
    'github_version': 'master/doc/' ,
}

# {%
#     if display_github %}
#         <li><a href="https://github.com/{{ github_user }}/{{ github_repo }}_
#             /tree/{{ github_version }}{{ conf_py_path }}{{ pagename }}.rst">
#             Show on GitHub</a></li>
#     {% endif %}

# The style sheet to use for HTML and HTML Help pages. A file of that name
# must exist either in Sphinx' static/ path, or in one of the custom paths
# given in html_static_path.
#html_style = 'default.css'
# html_theme = 'sphinxdoc'

# Read the docs style:
if os.environ.get('READTHEDOCS') != 'True':
    try:
        import sphinx_rtd_theme
    except ImportError:
        pass # assume we have sphinx >= 1.3
    else:
        html_theme_path = [sphinx_rtd_theme.get_html_theme_path()]
        html_theme = 'sphinx_rtd_theme'

def setup(app):
    app.add_stylesheet("fix_rtd.css")

# The name for this set of Sphinx documents. If None, it defaults to
# "<project> v<release> documentation".
#html_title = None

# If true, "Created using Sphinx" is shown in the HTML footer. Default is_
# ↪True.
#html_show_sphinx = False

# A shorter title for the navigation bar. Default is the same as html_title.
#html_short_title = None

# The name of an image file (within the static path) to place at the top of
```

(continues on next page)

(continued from previous page)

```

# the sidebar.
html_logo = 'images/theano_logo_allwhite_210x70.png'

# The name of an image file (within the static path) to use as favicon of the
# docs. This file should be a Windows icon file (.ico) being 16x16 or 32x32
# pixels large.
html_favicon = 'images/icon.ico'

# Add any paths that contain custom static files (such as style sheets) here,
# relative to this directory. They are copied after the builtin static files,
# so a file named "default.css" will overwrite the builtin "default.css".
html_static_path = ['images']

# If not '', a 'Last updated on:' timestamp is inserted at every page bottom,
# using the given strftime format.
html_last_updated_fmt = '%b %d, %Y'

# If true, SmartyPants will be used to convert quotes and dashes to
# typographically correct entities.
html_use_smartytags = True

# Custom sidebar templates, maps document names to template names.
html_sidebars = {}

# Additional templates that should be rendered to pages, maps page names to
# template names.
html_additional_pages = {}

# If false, no module index is generated.
html_use_modindex = True

# If false, no index is generated.
html_use_index = True

# If true, the index is split into individual pages for each letter.
html_split_index = False

# If true, the rest sources are included in the HTML build as _sources/<name>.
html_copy_source = True

# If true, an OpenSearch description file will be output, and all pages will
# contain a <link> tag referring to it. The value of this option must be the
# base URL from which the finished HTML is served.
html_use_opensearch = ''

# If nonempty, this is the file name suffix for HTML files (e.g. ".xhtml").
html_file_suffix = ''

# Output file base name for HTML help builder.
htmlhelp_basename = 'spnixgitdoc'

# Options for the linkcode extension

```

(continues on next page)

(continued from previous page)

```
# -----
# Resolve function
# This function is used to populate the (source) links in the API
def linkcode_resolve(domain, info):
    def find_source():
        # try to find the file and line number, based on code from numpy:
        # https://github.com/numpy/numpy/blob/master/doc/source/conf.py#L286
        obj = sys.modules[info['module']]
        for part in info['fullname'].split('.'):
            obj = getattr(obj, part)
        import inspect
        import os
        fn = inspect.getsourcefile(obj)
        fn = os.path.relpath(fn, start=os.path.dirname(theano.__file__))
        source, lineno = inspect.getsourcelines(obj)
        return fn, lineno, lineno + len(source) - 1

    if domain != 'py' or not info['module']:
        return None
    try:
        filename = '%s#L%d-L%d' % find_source()
    except Exception:
        filename = info['module'].replace('.', '/') + '.py'
    import subprocess
    # tag = subprocess.Popen(['git', 'rev-parse', 'HEAD'],
    #                       stdout=subprocess.PIPE,
    #                       universal_newlines=True).communicate()[0][-1]
    # https://github.com/runawayhorse001/statspy/blob/master/statspy/basics.py
    # https://github.com/runawayhorse001/SphinxGithub/blob/master/statspy/
    ↪basics.py
    return "https://github.com/runawayhorse001/SphinxGithub/blob/master/%s" %
    ↪(filename)

# Options for LaTeX output
# -----
latex_elements = {
    # The paper size ('letter' or 'a4').
    # latex_paper_size = 'a4',

    # The font size ('10pt', '11pt' or '12pt').
    'pointsize': '11pt',

    # Additional stuff for the LaTeX preamble.
    # latex_preamble = '',
}

# Grouping the document tree into LaTeX files. List of tuples
# (source start file, target name, title, author, document class
# [howto/manual]).
latex_documents = [
    ('index', 'sphinxgithub.tex', 'Sphinx with Github Webpages',
```

(continues on next page)

(continued from previous page)

(continues on next page)

(continued from previous page)

```
'\\def\\c{\\mathbf{c}}\\n'+\
'\\def\\W{\\mathbf{W}}\\n'+\
'\\def\\C{\\mathbf{C}}\\n'+\
'\\def\\P{\\mathbf{P}}\\n'+\
'\\def\\T{\\mathbf{T}}\\n'+\
'\\def\\B{\\mathcal{B}}\\n'+\
'\\def\\euler{\\mathrm{e}^{\\mathrm{i}\\pi} + 1 = 0}\\n'

# Documents to append as an appendix to all manuals.
latex_appendices = []

# If false, no module index is generated.
latex_use_modindex = True

default_role = 'math'
imgmath_divpng_args = ['-gamma 1.5', '-D 110']
#pngmath_divpng_args = ['-gamma', '1.5', '-D', '110', '-bg', 'Transparent']
imgmath_latex_preamble = '\\usepackage{amsmath}\\n'+\
    '\\usepackage{mathtools}\\n'+\
    '\\usepackage{amsfonts}\\n'+\
    '\\usepackage{amssymb}\\n'+\
    '\\usepackage{dsfont}\\n'+\
    '\\def\\Z{\\mathbb{Z}}\\n'+\
    '\\def\\R{\\mathbb{R}}\\n'+\
    '\\def\\bX{\\mathbf{X}}\\n'+\
    '\\def\\X{\\mathbf{X}}\\n'+\
    '\\def\\By{\\mathbf{y}}\\n'+\
    '\\def\\Bbeta{\\boldsymbol{\\beta}}\\n'+\
    '\\def\\U{\\mathbf{U}}\\n'+\
    '\\def\\V{\\mathbf{V}}\\n'+\
    '\\def\\V1{\\mathbf{1}}\\n'+\
    '\\def\\hU{\\hat{U}}\\n'+\
    '\\def\\hS{\\hat{\Sigma}}\\n'+\
    '\\def\\hV{\\hat{V}}\\n'+\
    '\\def\\E{\\mathbf{E}}\\n'+\
    '\\def\\F{\\mathbf{F}}\\n'+\
    '\\def\\x{\\mathbf{x}}\\n'+\
    '\\def\\h{\\mathbf{h}}\\n'+\
    '\\def\\v{\\mathbf{v}}\\n'+\
    '\\def\\nv{\\mathbf{v}^{{\\mathrm{\\bf -}}}}\\n'+\
    '\\def\\nh{\\mathbf{h}^{{\\mathrm{\\bf -}}}}\\n'+\
    '\\def\\s{\\mathbf{s}}\\n'+\
    '\\def\\b{\\mathbf{b}}\\n'+\
    '\\def\\c{\\mathbf{c}}\\n'+\
    '\\def\\W{\\mathbf{W}}\\n'+\
    '\\def\\C{\\mathbf{C}}\\n'+\
    '\\def\\P{\\mathbf{P}}\\n'+\
    '\\def\\T{\\mathbf{T}}\\n'+\
    '\\def\\B{\\mathcal{B}}\\n'+\
    '\\def\\euler{\\mathrm{e}^{\\mathrm{i}\\pi} + 1 = 0}\\n'
```

## 4.4 General Documentation Generator Configuration

### 4.4.1 Output Options

```

throot = os.path.abspath(
    os.path.join(sys.path[0], os.pardir, os.pardir))

options = defaultdict(bool)
opts, args = getopt.getopt(
    sys.argv[1:],
    'o:f:',
    ['rst', 'help', 'nopdf', 'cache', 'check', 'test'])
options.update(dict([x, y or True] for x, y in opts))
if options['--help']:
    print('Usage: %s [OPTIONS] [files...]' % sys.argv[0])
    print(' -o <dir>: output the html files in the specified dir')
    print(' --cache: use the doctree cache')
    print(' --rst: only compile the doc (requires sphinx)')
    print(' --nopdf: do not produce a PDF file from the doc, only HTML')
    print(' --test: run all the code samples in the documentaton')
    print(' --check: treat warnings as errors')
    print(' --help: this help')
    print('If one or more files are specified after the options then only '
        'those files will be built. Otherwise the whole tree is '
        'processed. Specifying files will implies --cache.')
    sys.exit(0)

if not(options['--rst'] or options['--test']):
    # Default is now rst
    options['--rst'] = True

```

### 4.4.2 Output Directory

```

def mkdir(path):
    try:
        os.mkdir(path)
    except OSError:
        pass

# create the putput folder docs, since github page will use /docs folder for
# Github page.
outdir = options['-o'] or (throot + '/docs')
# create the output folder latex
latexdir = options['-o'] or (throot + '/latex')

files = None
if len(args) != 0:
    files = [os.path.abspath(f) for f in args]
currentdir = os.getcwd()
mkdir(outdir)

```

(continues on next page)

(continued from previous page)

```
mkdir(latexdir)
os.chdir(outdir)
```

### 4.4.3 Documentation Compiler

```
def call_sphinx(builder, workdir):
    import sphinx
    if options['--check']:
        extraopts = ['-W']
    else:
        extraopts = []
    if not options['--cache'] and files is None:
        extraopts.append('-E')
    docpath = os.path.join(throot, 'doc')
    inopt = [docpath, workdir]
    if files is not None:
        inopt.extend(files)
    ret = sphinx.build_main(['', '-b', builder] + extraopts + inopt)
    if ret != 0:
        sys.exit(ret)

    if options['--all'] or options['--rst']:
        mkdir("doc")
        sys.path[0:0] = [os.path.join(throot, 'doc')]
        call_sphinx('html', '.')

        if not options['--nopdf']:
            # Generate latex file in a temp directory
            import tempfile
            #workdir = tempfile.mkdtemp()
            workdir = latexdir
            call_sphinx('latex', workdir)
            # Compile to PDF
            os.chdir(workdir)
            os.system('make')
            try:
                shutil.copy(os.path.join(workdir, 'sphinxygithub.pdf'), outdir)
                os.chdir(outdir)
                # remove the workdir folder
                #shutil.rmtree(workdir)
            except OSError as e:
                print('OSError:', e)
            except IOError as e:
                print('IOError:', e)

    if options['--test']:
        mkdir("doc")
```

(continues on next page)

(continued from previous page)

```

sys.path[0:0] = [os.path.join(throot, 'doc')]
call_sphinx('doctest', '.')

# To go back to the original current directory.
os.chdir(currentdir)

# Reset THEANO_FLAGS
# os.environ['THEANO_FLAGS'] = env_th_flags

```

#### 4.4.4 Makefile Wrapper

```

all:
    python scripts/docgen.py

```

### 4.5 Full docgen.py Script

```

#####
# I heavily borrowed, modified and used the configuration in docgen.py of_
# →Theano
# package project. I will keep all the comments from Theano team and the
# copyright of this file belongs to Theano team.
# reference:
#
# Theano repository: https://github.com/Theano/Theano
# docgen.py: https://github.com/Theano/Theano/blob/master/doc/scripts/docgen.
# →py
#####
from __future__ import print_function
import sys
import os
import shutil
import inspect
import getopt
from collections import defaultdict

if __name__ == '__main__':
    throot = os.path.abspath(
        os.path.join(sys.path[0], os.pardir, os.pardir))

    options = defaultdict(bool)
    opts, args = getopt.getopt(
        sys.argv[1:],
        'o:f:',
        ['rst', 'help', 'nopdf', 'cache', 'check', 'test'])
    options.update(dict([x, y or True] for x, y in opts))
    if options['--help']:

```

(continues on next page)

(continued from previous page)

```

print('Usage: %s [OPTIONS] [files...]' % sys.argv[0])
print(' -o <dir>: output the html files in the specified dir')
print(' --cache: use the doctree cache')
print(' --rst: only compile the doc (requires sphinx)')
print(' --nopdf: do not produce a PDF file from the doc, only HTML')
print(' --test: run all the code samples in the documentation')
print(' --check: treat warnings as errors')
print(' --help: this help')
print('If one or more files are specified after the options then only
→'
      'those files will be built. Otherwise the whole tree is '
      'processed. Specifying files will implies --cache.')
sys.exit(0)

if not(options['--rst'] or options['--test']):
    # Default is now rst
    options['--rst'] = True

def mkdir(path):
    try:
        os.mkdir(path)
    except OSError:
        pass

# create the output folder docs, since github page will use /docs folder
→for Github page.
outdir = options['-o'] or (throot + '/docs')
# create the output folder latex
latexdir = options['-o'] or (throot + '/latex')

files = None
if len(args) != 0:
    files = [os.path.abspath(f) for f in args]
currentdir = os.getcwd()
mkdir(outdir)
mkdir(latexdir)
os.chdir(outdir)

# add .gitignore file to your github repository
ignore_path = os.path.join(throot, '.gitignore')
if not os.path.exists(ignore_path):
    ignore_txt = open(throot + '/doc/scripts/+' + 'gitignore.txt')
    gitignore = open(ignore_path, 'a')
    for x in ignore_txt.readlines():
        gitignore.write(x)
    ignore_txt.close()
    gitignore.close()

# add .nojekyll file to fix the github pages issues
nojekyll_path = os.path.join(outdir, '.nojekyll')
if not os.path.exists(nojekyll_path):
    nojekyll = open(nojekyll_path, 'a')

```

(continues on next page)

(continued from previous page)

```

nojekyll.close()

# Make sure the appropriate 'theano' directory is in the PYTHONPATH
pythonpath = os.environ.get('PYTHONPATH', '')
pythonpath = os.pathsep.join([throot, pythonpath])
sys.path[0:0] = [throot] # We must not use os.environ.

# Make sure we don't use gpu to compile documentation
env_th_flags = os.environ.get('THEANO_FLAGS', '')
os.environ['THEANO_FLAGS'] = 'device=cpu,force_device=True'

def call_sphinx(builder, workdir):
    import sphinx
    if options['--check']:
        extraopts = ['-W']
    else:
        extraopts = []
    if not options['--cache'] and files is None:
        extraopts.append('-E')
    docpath = os.path.join(throot, 'doc')
    inopt = [docpath, workdir]
    if files is not None:
        inopt.extend(files)
    ret = sphinx.build_main(['', '-b', builder] + extraopts + inopt)
    if ret != 0:
        sys.exit(ret)

    if options['--all'] or options['--rst']:
        mkdir("doc")
        sys.path[0:0] = [os.path.join(throot, 'doc')]
        call_sphinx('html', '.')

        if not options['--nopdf']:
            # Generate latex file in a temp directory
            import tempfile
            #workdir = tempfile.mkdtemp()
            workdir = latexdir
            call_sphinx('latex', workdir)
            # Compile to PDF
            os.chdir(workdir)
            os.system('make')
            try:
                shutil.copy(os.path.join(workdir, 'sphinxygithub.pdf'), outdir)
                os.chdir(outdir)
                # remove the workdir folder
                #shutil.rmtree(workdir)
            except OSError as e:
                print('OSError:', e)
            except IOError as e:
                print('IOError:', e)

```

(continues on next page)

(continued from previous page)

```
if options['--test']:
    mkdir("doc")
    sys.path[0:0] = [os.path.join(throot, 'doc')]
    call_sphinx('doctest', '.')

# To go back to the original current directory.
os.chdir(currentdir)

# Reset THEANO_FLAGS
#os.environ['THEANO_FLAGS'] = env_th_flags
```

---

CHAPTER  
FIVE

---

## RESTRUCTUREDTEXT MARKUP

---

### Chinese proverb

Making full preparation will not delay your job but quicken the process. – old Chinese proverb

---

## 5.1 reStructuredText Primer

I would refer the reader to [Sphinx2019] and [Georg2018] for more details.

### 5.1.1 Sections

Sections are identified through their titles, which are marked up with adornment: “underlines” below the title text, or underlines and matching “overlines” above the title. More details can be found at: <http://docutils.sourceforge.net/docs/ref/rst/restructuredtext.html#sections>

reStructuredText:

```
=====
Section Title
=====

Subsection Title
++++++

Subsubsection Title
-----
```

### 5.1.2 Paragraphs

The paragraph is the most basic block in a reST document. Paragraphs are simply chunks of text separated by one or more blank lines. As in Python, indentation is significant in reST, so all lines of the same paragraph must be left-aligned to the **same level of indentation**. More details can be found at: <http://docutils.sourceforge.net/docs/ref/rst/restructuredtext.html#paragraphs>

## 1. General Paragraphs

reStructuredText:

```
This is the first demo paragraph. The blank line above  
the first line is required; The blank line below the last  
line is required.
```

```
This is the second demo paragraph. The blank line above  
the first line is required; The blank line below the last  
line is required.
```

Syntax diagram:

```
+-----+  
| paragraph |  
|           |  
+-----+  
  
+-----+  
| paragraph |  
|           |  
+-----+
```

Result:

This is the first demo paragraph. The blank line above the first line is required; The blank line below the last line is required.

This is the second demo paragraph. The blank line above the first line is required; The blank line below the last line is required.

## 2. Bullet Lists Paragraphs

reStructuredText:

```
- This is the first bullet list item. The blank line above the  
first list item is required; blank lines between list items  
(such as below this paragraph) are optional.  
  
- This is the first paragraph in the second item in the list.  
  
    This is the second paragraph in the second item in the list.  
    The blank line above this paragraph is required. The left edge  
    of this paragraph lines up with the paragraph above, both  
    indented relative to the bullet.  
  
    - This is a sublist. The bullet lines up with the left edge of  
      the text blocks above. A sublist is a new list so requires a  
      blank line above and below.  
  
    - This is a sublist. The bullet lines up with the left edge of  
      the text blocks above. A sublist is a new list so requires a  
      blank line above and below.
```

(continues on next page)

(continued from previous page)

```
- This is the third item of the main list.
```

```
This paragraph is not part of the list.
```

Syntax diagram:

```
+-----+-----+
| "- " | list item           |
+-----| (body elements) +   |
+-----+-----+
```

Result:

- This is the first bullet list item. The blank line above the first list item is required; blank lines between list items (such as below this paragraph) are optional.
- This is the first paragraph in the second item in the list.

This is the second paragraph in the second item in the list. The blank line above this paragraph is required. The left edge of this paragraph lines up with the paragraph above, both indented relative to the bullet.

- This is a sublist. The bullet lines up with the left edge of the text blocks above. A sublist is a new list so requires a blank line above and below.
- This is a sublist. The bullet lines up with the left edge of the text blocks above. A sublist is a new list so requires a blank line above and below.
- This is the third item of the main list.

This paragraph is not part of the list.

### 3. Bullet Lists Paragraphs

reStructuredText:

```
1. This is the first bullet list item. The blank line above the
first list item is required; blank lines between list items
(such as below this paragraph) are optional.
```

```
2. This is the first paragraph in the second item in the list.
```

This is the second paragraph in the second item in the list. The blank line above this paragraph is required. The left edge of this paragraph lines up with the paragraph above, both indented relative to the bullet.

a. This is a sublist. The bullet lines up with the left edge of the text blocks above. A sublist is a new list so requires a blank line above and below.

b. This is a sublist. The bullet lines up with the left edge of

(continues on next page)

(continued from previous page)

```
the text blocks above. A sublist is a new list so requires a  
blank line above and below.
```

```
3. This is the third item of the main list.
```

Result:

1. This is the first bullet list item. The blank line above the first list item is required; blank lines between list items (such as below this paragraph) are optional.
2. This is the first paragraph in the second item in the list.

This is the second paragraph in the second item in the list. The blank line above this paragraph is required. The left edge of this paragraph lines up with the paragraph above, both indented relative to the bullet.

- a. **This is a sublist. The bullet lines up with the left edge of** the text blocks above. A sublist is a new list so requires a blank line above and below.
- b. **This is a sublist. The bullet lines up with the left edge of** the text blocks above. A sublist is a new list so requires a blank line above and below.

3. This is the third item of the main list.

Syntax diagram:

```
+-----+-----+  
| "1. " | list item |  
+-----| (body elements) + |  
+-----+
```

Result:

1. This is the first bullet list item. The blank line above the first list item is required; blank lines between list items (such as below this paragraph) are optional.
2. This is the first paragraph in the second item in the list.

This is the second paragraph in the second item in the list. The blank line above this paragraph is required. The left edge of this paragraph lines up with the paragraph above, both indented relative to the bullet.

- a. This is a sublist. The bullet lines up with the left edge of the text blocks above. A sublist is a new list so requires a blank line above and below.

- b. This is a sublist. The bullet lines up with the left edge of the text blocks above. A sublist is a new list so requires a blank line above and below.

3. This is the third item of the main list.

#### 4. Blocked Paragraphs

##### a. Line Blocks

reStructuredText:

```
Take it away, Eric the Orchestra Leader!
```

```
| A one, two, a one two three four  
|  
| Half a bee, philosophically,  
|     must, *ipso facto*, half not be.  
| But half the bee has got to be,  
|     *vis a vis* its entity. D'you see?  
  
| But can a bee be said to be  
|     or not to be an entire bee,  
|         when half the bee is not a bee,  
|             due to some ancient injury?  
  
| Singing...
```

Syntax diagram:

```
+-----+  
| " | " | line  
+-----| continuation line  
      +-----+
```

Result:

```
Take it away, Eric the Orchestra Leader!
```

```
A one, two, a one two three four
```

```
Half a bee, philosophically,  
    must, ipso facto, half not be.
```

```
But half the bee has got to be,  
    vis a vis its entity. D'you see?
```

```
But can a bee be said to be  
    or not to be an entire bee,  
        when half the bee is not a bee,  
            due to some ancient injury?
```

```
Singing...
```

## 5. Doctest Blocks

reStructuredText:

```
This is an ordinary paragraph.  
  
>>> print 'this is a Doctest block'  
this is a Doctest block
```

(continues on next page)

(continued from previous page)

The following is a literal block::

```
>>> This is not recognized as a doctest block by  
reStructuredText. It *will* be recognized by the doctest  
module, though!
```

Result:

This is an ordinary paragraph.

```
>>> print 'this is a Doctest block'  
this is a Doctest block
```

The following is a literal block:

```
>>> This is not recognized as a doctest block by  
reStructuredText. It *will* be recognized by the doctest  
module, though!
```

### 3. Field Lists

reStructuredText:

```
:Date: 2001-08-16  
:Version: 1  
:Authors: - Me  
          - Myself  
          - I  
:Indentation: Since the field marker may be quite long, the second  
and subsequent lines of the field body do not have to line up  
with the first line, but they must be indented relative to the  
field name marker, and they must line up with each other.  
:Parameter i: integer
```

Result:

**Date** 2001-08-16

**Version** 1

**Authors**

- Me
- Myself
- I

**Indentation** Since the field marker may be quite long, the second and subsequent lines of the field body do not have to line up with the first line, but they must be indented relative to the field name marker, and they must line up with each other.

**Parameter i** integer

### 5.1.3 Table

More details can be found at: <http://docutils.sourceforge.net/docs/ref/rst/restructuredtext.html#grid-tables>

#### 1. Grid Tables

reStructuredText:

Header row, column 1 (header rows optional)	Header 2	Header 3	Header 4
body row 1, column 1	column 2	column 3	column 4
body row 2	Cells may span columns.		
body row 3	Cells may span rows.	<ul style="list-style-type: none"> <li>– Table cells</li> <li>– contain</li> <li>– body elements.</li> </ul>	
body row 4			

Result:

Header row, column 1 (header rows optional)	Header 2	Header 3	Header 4
body row 1, column 1	column 2	column 3	column 4
body row 2	Cells may span columns.		
body row 3	Cells may span rows.	<ul style="list-style-type: none"> <li>– Table cells</li> <li>– contain</li> <li>– body elements.</li> </ul>	
body row 4			

#### 2. Simple Tables

reStructuredText:

A	B	A and B
False	False	False
True	False	False
False	True	False
True	True	True

Result:

A	B	A and B
False	False	False
True	False	False
False	True	False
True	True	True

reStructuredText:

```
===== =====
col 1 col 2
===== =====
1     Second column of row 1.
2     Second column of row 2.
      Second line of paragraph.
3     - Second column of row 3.

      - Second item in bullet
        list (row 3, column 2).
\    Row 4; column 1 will be empty.
===== =====
```

Result:

col 1	col 2
1	Second column of row 1.
2	Second column of row 2. Second line of paragraph.
3	<ul style="list-style-type: none"><li>• Second column of row 3.</li><li>• Second item in bullet list (row 3, column 2).</li></ul>
Row 4; column 1 will be empty.	

### 3. CSV Tables

reStructuredText:

```
.. csv-table:: Frozen Delights!
:header: "Treat", "Quantity", "Description"
:widths: 15, 10, 30

"Albatross", 2.99, "On a stick!"
"Crunchy Frog", 1.49, "If we took the bones out, it wouldn't be
crunchy, now would it?"
"Gannet Ripple", 1.99, "On a stick!"
```

Table 1: Frozen Delights!

Treat	Quantity	Description
Albatross	2.99	On a stick!
Crunchy Frog	1.49	If we took the bones out, it wouldn't be crunchy, now would it?
Gannet Ripple	1.99	On a stick!

### 4. List Tables

reStructuredText:

```
.. list-table:: Frozen Delights!
:widths: 15 10 30
:header-rows: 1

* - Treat
  - Quantity
  - Description
* - Albatross
  - 2.99
  - On a stick!
* - Crunchy Frog
  - 1.49
  - If we took the bones out, it wouldn't be
    crunchy, now would it?
* - Gannet Ripple
  - 1.99
  - On a stick!
```

Table 2: Frozen Delights!

Treat	Quantity	Description
Albatross	2.99	On a stick!
Crunchy Frog	1.49	If we took the bones out, it wouldn't be crunchy, now would it?
Gannet Ripple	1.99	On a stick!

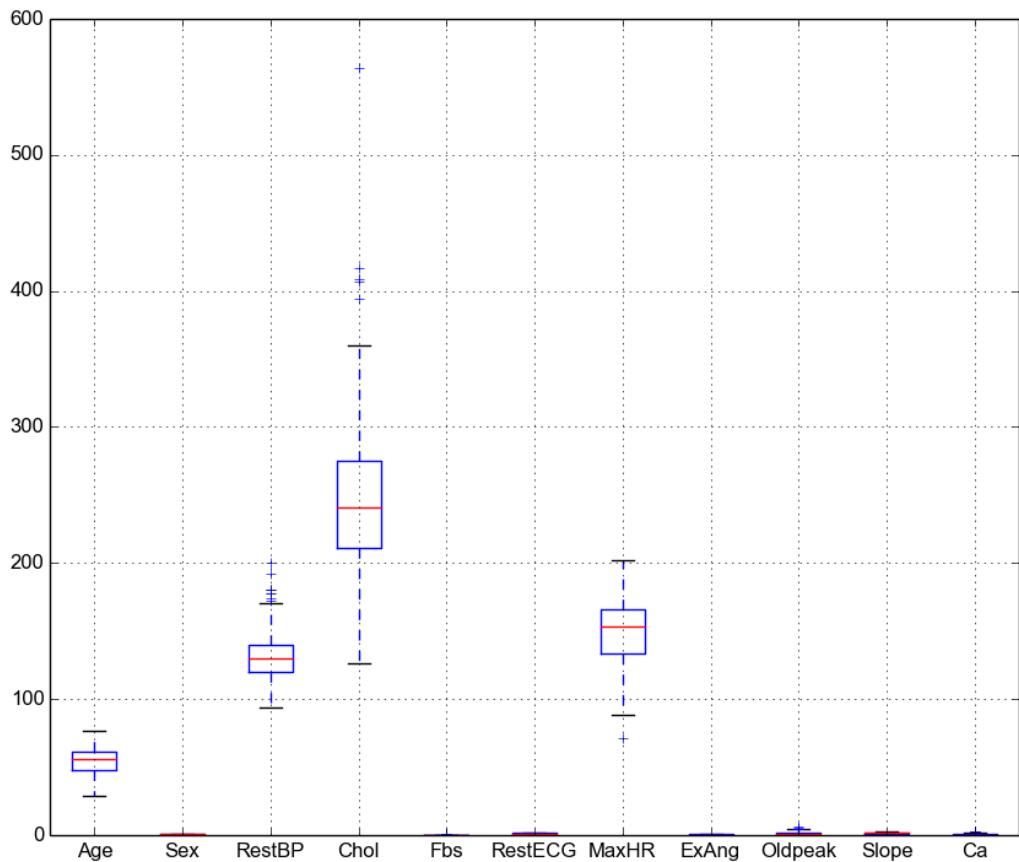
### 5.1.4 Images and Figures

There are two image directives: `image` and `figure`. More details can be found at: <http://docutils.sourceforge.net/docs/ref/rst/directives.html#image>.

#### 1. Simple import

reStructuredText:

```
.. image:: images/boxp.png
```



reStructuredText:

```
.. figure:: images/avg_rating_mon.png
:scale: 50 %
:alt: map to buried treasure
```

This is the caption of the figure (a simple paragraph).

## 2. Complex import

reStructuredText:

```
.. figure:: images/bboxp.png
:height: 400 px
:width: 800 px
:scale: 50 %
:alt: alternate text
:align: right
```

This is the caption of the figure (a simple paragraph).

(continues on next page)

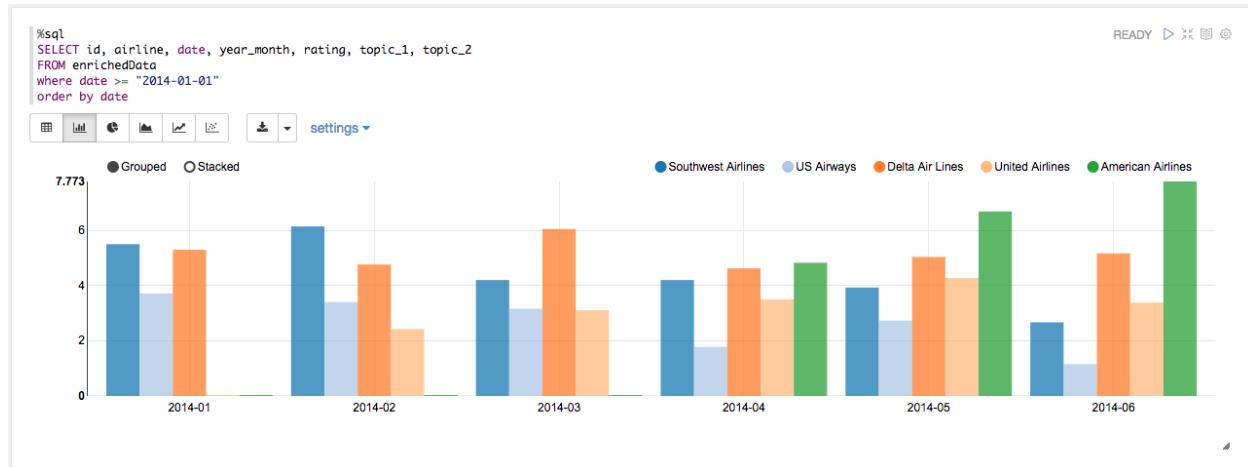


Fig. 1: This is the caption of the figure (a simple paragraph).

(continued from previous page)

Images and Figures  
↪Images and Figures  
Images and Figures Images and Figures Images and Figures Images and Figures Images and Figures  
↪Images and Figures  
Images and Figures Images and Figures Images and Figures Images and Figures Images and Figures  
↪Images and Figures  
Images and Figures Images and Figures Images and Figures Images and Figures Images and Figures  
↪Images and Figures

```
.. figure:: images/boxp.png
:height: 400 px
:width: 800 px
:scale: 50 %
:alt: alternate text
:align: center
```

This is the caption of the figure (a simple paragraph).

Images and Figures  
↪ Images and Figures  
Images and Figures Images and Figures Images and Figures Images and Figures Images and Figures  
↪ Images and Figures  
Images and Figures Images and Figures Images and Figures Images and Figures Images and Figures  
↪ Images and Figures  
Images and Figures Images and Figures Images and Figures Images and Figures Images and Figures  
↪ Images and Figures

```
.. figure:: images/bboxp.png
   :height: 400 px
   :width: 800 px
   :scale: 50 %
   :alt: alternate text
```

(continues on next page)

(continued from previous page)

:align: left

This is the caption of the figure (a simple paragraph).

Images and Figures Images and Figures Images and Figures Images and Figures  
↳ Images and Figures  
Images and Figures Images and Figures Images and Figures Images and Figures  
↳ Images and Figures  
Images and Figures Images and Figures Images and Figures Images and Figures  
↳ Images and Figures  
Images and Figures Images and Figures Images and Figures Images and Figures  
↳ Images and Figures

### Result:

# Images and Fig- ures

# Im- ages and Fig- ures

# Im- ages and Fig- ures

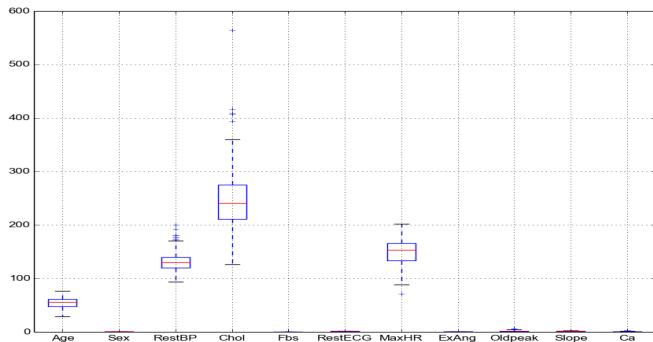


Fig. 2: This is the caption of the figure (a simple paragraph).

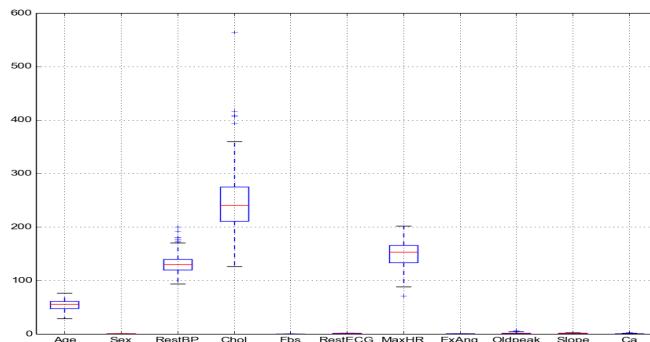


Fig. 3: This is the caption of the figure (a simple paragraph).

Images and Figures  
Images and Figures Images and Figures Images and Figures Images and Figures Images and Figures  
Images and Figures Images and Figures Images and Figures Images and Figures Images and Figures  
Images and Figures Images and Figures Images and Figures Images and Figures Images and Figures

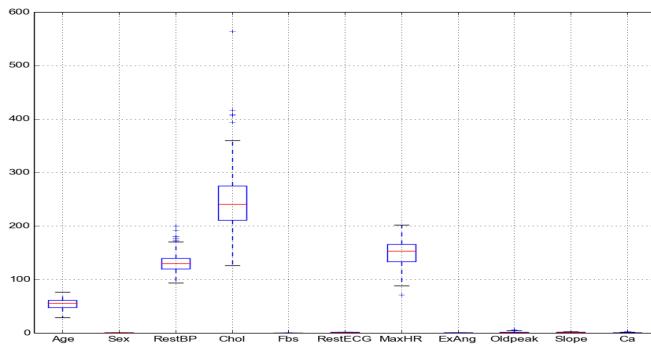


Fig. 4: This is the caption of the figure (a simple paragraph).

### 3. Figures in table

## reStructuredText:

Figures	Description
.. figure:: images/corr.png	Figure 1: test
:scale: 20 %	
.. image:: images/corr.png	Figure 2: test
:scale: 20 %	
.. image:: images/corr.png	Figure 3: test
:scale: 20 %	

## Result:

Figures	Description
	Figure 1: test
	Figure 2: test
	Figure 3: test

## 5.1.5 Math

The `math` role marks its content as mathematical notation (inline formula). More details can be found at: <http://www.sphinx-doc.org/es/stable/ext/math.html>.

The input language for mathematics is LaTeX markup. I will not do a LaTex tutorial as here.

## 1. Inline formula

reStructuredText:

```
The area of a circle is :math:`A_{\text{c}} = (\pi/4) d^2`.
```

RResult:

The area of a circle is  $A_c = (\pi/4)d^2$ .

## 2. Equations

reStructuredText:

```
.. math::
:label: eq_lin_cost_func

\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \| \hat{X} \beta - \hat{Y} \|^2
```

The equation :eq:`eq\_lin\_cost\_func` is the cost function for linear regression.

Result:

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \| \hat{X} \beta - \hat{Y} \|^2 \quad (5.1)$$

The equation (5.1) is the cost function for linear regression.

## 3. User defined symbol and equation

Add your definitions to the `latex_elements['preamble']` and `imgmath_latex_preamble`, then you can apply your own notations for symbol and equations.

my definitions for  $\mathcal{B}$  symbol and  $e^{i\pi} + 1 = 0$  equation:

```
'\\def\\B{{\\bf \\mathcal B}}\n'+ \
'\\def\\euler{\\ e^{i\\pi} + 1 = 0}\n'
```

reStructuredText:

```
The is a test for the user defined math symbol: :math:`\B`.
```

The is a test for the user defined math equation:

```
.. math::  
  
\\euler
```

The is a test for the user defined math symbol:  $\mathcal{B}$ .

The is a test for the user defined math equation:

$$e^{i\pi} + 1 = 0$$

#### 4. More examples

reStructuredText:

```
.. math::  
  
    f(x)  
    =  
    \Biggl \lbrace  
    {  
    0, \text{ if }  
    { x >0 }  
    \atop  
    1 \text{ otherwise }  
    }  
  
.. math::  
  
    (a + b)^2 &= (a + b)(a + b) \\  
    &= a^2 + 2ab + b^2  
  
.. math::  
:nowrap:  
  
    \begin{eqnarray}  
    y &= ax^2 + bx + c \nonumber\\  
    f(x) &= x^2 + 2xy + y^2 \nonumber  
    \end{eqnarray}  
  
.. math:: e^{i\pi} + 1 = 0  
:label: euler
```

Result:

$$f(x) = \begin{cases} 0, & \text{if } x > 0 \\ 1 & \text{otherwise} \end{cases}$$

$$\begin{aligned} (a + b)^2 &= (a + b)(a + b) \\ &= a^2 + 2ab + b^2 \end{aligned}$$

$$\begin{aligned} y &= ax^2 + bx + c \\ f(x) &= x^2 + 2xy + y^2 \end{aligned}$$

$$e^{i\pi} + 1 = 0 \tag{5.2}$$

#### 5.1.6 Source Codes

##### 1. Source code block

reStructuredText:

```
.. code-block:: python

    """
    This is a source Python code demo for Sphinx.
    @date: Apr 25, 2016
    @author: Wenqiang Feng
    """

    import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    from pandas.tools.plotting import scatter_matrix
    from docutils.parsers.rst.directives import path

    if __name__ == '__main__':
        print("This is a source Python code demo for Sphinx.")
```

Result:

```
"""
This is a source Python code demo for Sphinx.
@date: Apr 25, 2016
@author: Wenqiang Feng
"""

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from pandas.tools.plotting import scatter_matrix
from docutils.parsers.rst.directives import path

if __name__ == '__main__':
    print("This is a source Python code demo for Sphinx.")
```

reStructuredText:

```
.. code-block:: r

    """
    This is a source R code demo for Sphinx.
    @date: Apr 25, 2016
    @author: Wenqiang Feng
    """

    library(reshape2)
    library(ggplot2)

    # import data
    d <- melt(diamonds[,-c(2:4)])
    # plot histogram
```

(continues on next page)

(continued from previous page)

```
ggplot(d,aes(x = value)) +  
    facet_wrap(~variable,scales = "free_x") +  
    geom_histogram()  
  
print("This is a source R code demo for Sphinx.")
```

```
'''  
This is a source R code demo for Sphinx.  
@date: Apr 25, 2016  
@author: Wengiang Feng  
library(reshape2)  
library(ggplot2)  
  
# import data  
d <- melt(diamonds[,-c(2:4)])  
# plot histogram  
ggplot(d,aes(x = value)) +  
    facet_wrap(~variable,scales = "free_x") +  
    geom_histogram()  
  
print("This is a source R code demo for Sphinx.")
```

## 2. Source code import

- Python Source code

reStructuredText:

```
.. literalinclude:: code/sourceCodePy.py  
    :language: python
```

Result:

```
'''  
This is a source Python code demo for Sphinx.  
@date: Apr 25, 2016  
@author: Wengiang Feng  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
from pandas.tools.plotting import scatter_matrix  
from docutils.parsers.rst.directives import path  
  
if __name__ == '__main__':  
  
    print("This is a source Python code demo for Sphinx.")
```

- R Source code

reStructuredText:

```
.. literalinclude:: code/sourceCodeR.R
    :language: r
```

Result:

```
'''
This is a source R code demo for Sphinx.
@date: Apr 25, 2016
@author: Wenqiang Feng
'''

library(reshape2)
library(ggplot2)

# import data
d <- melt(diamonds[,-c(2:4)])
# plot histogram
ggplot(d,aes(x = value)) +
  facet_wrap(~variable,scales = "free_x") +
  geom_histogram()

print("This is a source R code demo for Sphinx.")
```

## 5.1.7 Reference

### 1. Paper reference

reStructuredText:

```
.. [Ref] Book or article reference, URL or whatever.

Lorem ipsum [Ref] dolor sit amet.
```

Result:

Lorem ipsum [Ref] dolor sit amet.

I would refer the reader to [Sphinx2019] for more details.

### 2. Equation reference

reStructuredText:

```
.. math::
:label: eq_condition

f(x)
=
\Biggl \lbrace
{
0,\text{ if }
```

(continues on next page)

(continued from previous page)

```
{ x >0}  
\atop  
1 \text{ otherwise }
```

The Equation :eq:`eq\_condition` is the definition of :math:`f(x)`.

Result:

$$f(x) = \begin{cases} 0, & \text{if } x > 0 \\ 1 & \text{otherwise} \end{cases} \quad (5.3)$$

The Equation (5.3) is the definition of  $f(x)$ .

## 4. Figure reference

reStructuredText:

```
... _fig_hist_demo:  
.. figure:: images/avg_rating_mon.png  
    :scale: 50 %  
    :alt: map to buried treasure
```

The histogram of the grouped dataset

The Figure. :ref:`fig\_hist\_demo` is the histogram of the grouped dataset.

Result:

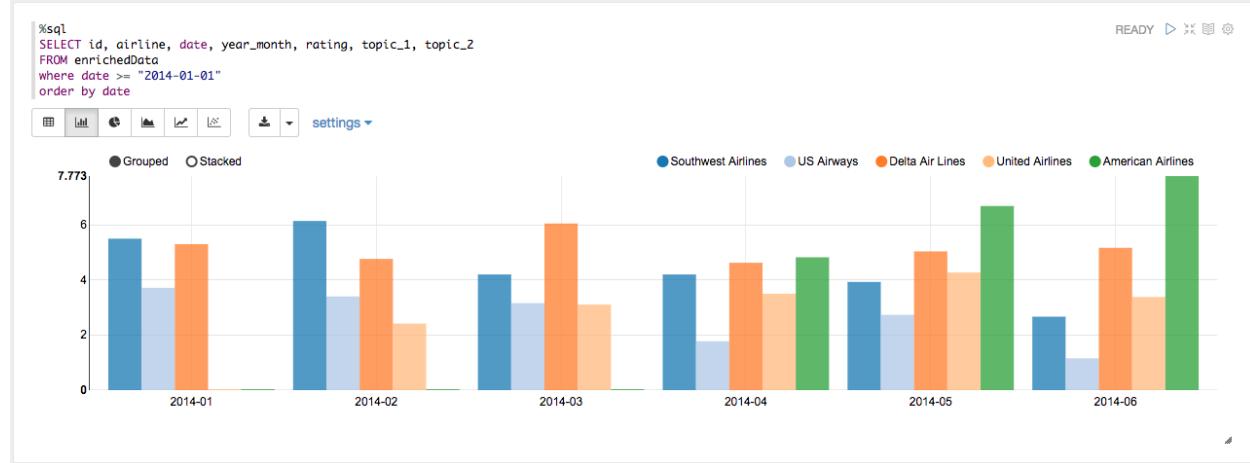


Fig. 5: The histogram of the grouped dataset

The Figure. *The histogram of the grouped dataset* is the histogram of the grouped dataset.

## 5. Table reference

reStructuredText:

```
.. _table_demo:
.. table:: The general table demo

+-----+-----+-----+-----+
| Header row, column 1 | Header 2 | Header 3 | Header 4 |
| (header rows optional) |           |           |           |
+=====+=====+=====+=====+
| body row 1, column 1 | column 2 | column 3 | column 4 |
+-----+-----+-----+-----+
| body row 2           | Cells may span columns. |
+-----+-----+-----+-----+
| body row 3           | Cells may   | - Table cells   |
+-----+-----+-----+-----+
| body row 4           | span rows. | - contain      |
|                       |           | - body elements. |
+-----+-----+-----+-----+
```

Please see the above Table. :ref:`table\_demo`.

Result:

Table 3: The general table demo

Header row, column 1 (header rows optional)	Header 2	Header 3	Header 4
body row 1, column 1	column 2	column 3	column 4
body row 2	Cells may span columns.		
body row 3	Cells may span rows.	<ul style="list-style-type: none"> <li>• Table cells</li> <li>• contain</li> <li>• body elements.</li> </ul>	
body row 4			

Please see the above Table. *The general table demo*.

## 6. Footnotes

reStructuredText:

```
 Lorem ipsum [#f1]_ dolor sit amet ... [#f2]_
 .. rubric:: Footnotes
 .. [#f1] Text of the first footnote.
 .. [#f2] Text of the second footnote.
```

Result:

Lorem ipsum<sup>1</sup> dolor sit amet ...<sup>2</sup>

## 7. Hyperlinks

---

<sup>1</sup> Text of the first footnote.

<sup>2</sup> Text of the second footnote.

- General hyperlink

reStructuredText:

```
You are more than welcome to visit my personal webpage: `Feng Website`_.  
.. _Feng Website: http://web.utk.edu/~wfeng1/
```

Result:

You are more than welcome to visit my personal webpage: Feng Website.

- Embedded Youtube link:

reStructuredText:

```
.. raw:: html  
  
<iframe width="700" height="315"  
src="https://www.youtube.com/embed/mrEee4bhc7Y"  
frameborder="0" allowfullscreen></iframe>
```

Result:

```
Warning: You have to use the hyperlink: https://www.youtube.com/embed/ + name.
```

## 5.2 Roles

A role or “custom interpreted text role” is an inline piece of explicit markup. It signifies that the enclosed text should be interpreted in a specific way. Sphinx uses this to provide semantic markup and cross-referencing of identifiers, as described in the appropriate section. More details can be found at: <http://docutils.sourceforge.net/docs/ref/rst/roles.html#customization>

### 5.2.1 Standard Roles

- Line markup

reStructuredText:

```
* emphasis - equivalent of *emphasis*  
  
* strong - equivalent of **strong**  
  
* literal - equivalent of ``literal``  
  
* subscript - H\ :sub:`2`\ O  
  
* superscript - E = mc\ :sup:`2`
```

(continues on next page)

(continued from previous page)

```
* title-reference – for titles of books, periodicals, and other materials
```

Result:

- emphasis – equivalent of *emphasis*
- strong – equivalent of **strong**
- literal – equivalent of `literal`
- subscript –  $H_2O$
- superscript –  $E = mc^2$
- title-reference – for titles of books, periodicals, and other materials

## 5.2.2 Specialized Roles

- `raw`

reStructuredText:

```
.. raw:: html

<iframe width="700" height="315"
src="https://www.youtube.com/embed/2Mg8QD0F1dQ"
frameborder="0" allowfullscreen></iframe>
```

Result:

reStructuredText:

```
.. role:: raw-html(raw)
:format: html

If there just *has* to be a line break here,
:raw-html:`<br />`
it can be accomplished with a "raw"-derived role.
But the line block syntax should be considered first.
```

Result:

If there just *has* to be a line break here,

it can be accomplished with a “raw”-derived role. But the line block syntax should be considered first.

- `replace`

reStructuredText:

```
.. |sphx| replace:: Sphinx
.. |reST| replace:: reStructuredText
```

(continues on next page)

(continued from previous page)

```
|reST| is awesome!
```

Sphinx and reStructuredText are awesome!

## 5.3 Directives

A directive is a generic block of explicit markup. Along with roles, it is one of the extension mechanisms of reST, and Sphinx makes heavy use of it.

### 5.3.1 Admonitions

Admonitions: attention, caution, danger, error, hint, important, note, tip, warning

- **attention**

reStructuredText:

```
.. attention::
```

```
    You neen to pay attention at here!
```

Result:

```
Attention: You neen to pay attention at here!
```

- **caution**

reStructuredText:

```
.. caution::
```

```
    This is a caution alert!
```

Restlut:

```
Caution: This is a caution alert!
```

- **important**

reStructuredText:

```
.. important::
```

```
    This is important!
```

Result:

---

**Important:** This is important!

---

- **User defined admonition**

reStructuredText:

```
.. admonition:: User defined name  
    You can make up your own admonition too.
```

Result:

---

### User defined name

You can make up your own admonition too.

---

- **seealso**

reStructuredText:

```
.. seealso::  
  
    The authoritative `reStructuredText User Documentation  
    <http://docutils.sourceforge.net/rst.html>`_. The "ref" links in this  
    document link to the description of the individual constructs in the reST  
    reference.
```

Result:

**See also:**

The authoritative [reStructuredText User Documentation](#). The “ref” links in this document link to the description of the individual constructs in the reST reference.

See more details at [Admonitions](#).



---

# CHAPTER SIX

---

## API EXAMPLE

### 6.1 rnorm

`statspy.basics.rnorm(n, mean=0, sd=1)`

Random generation for the normal distribution with mean equal to mean and standard deviation equation to sd same functions as rnorm in r: `rnorm(n, mean=0, sd=1)`

#### Parameters

- **n** – the number of the observations
- **mean** – vector of means
- **sd** – vector of standard deviations

**Returns** the vector of the random numbers

**Author** Wenqiang Feng

**Email** von198@gmail.com

### 6.2 dnorm

`statspy.basics.dnorm(x, mean=0, sd=1, log=False)`

Density of the normal distribution with mean equal to mean and standard deviation equation to sd same functions as rnorm in r: `dnorm(x, mean=0, sd=1, log=FALSE)`

#### Parameters

- **x** – the vector od quantiles
- **mean** – vector of means
- **sd** – vector of standard deviations

**Returns** the list of the density

**Author** Wenqiang Feng

**Email** von198@gmail.com

## 6.3 runif

`statspy.basics.runif(n, min=0, max=1)`

Random generation from the uniform distribution same functions as rnorm in r: `runif(n, min=0, max=1)`

### Parameters

- **n** – the number of the observations
- **min** – the lower limit of the distribution
- **max** – the upper limit of the distribution

**Returns** the list of n uniform random numbers

**Author** Wenqiang Feng

**Email** von198@gmail.com

## 6.4 T-test

`statspy.tests.t_test(x, y=None, mu=0.0, conf_level=0.95)`

Performs one and two sample t-tests on vectors of data.

same functions as t.test in r: `t.test(x, ...)`

```
t.test(x, y = NULL,  
       alternative = c("two.sided", "less", "greater"),  
       mu = 0, paired = FALSE, var.equal = FALSE,  
       conf.level = 0.95, ...)
```

### Parameters

- **x** – a (non-empty) numeric vector of data values.
- **y** – an optional (non-empty) numeric vector of data values.
- **mu** – vector of standard deviations.
- **conf\_level** – confidence level of the interval.

**Returns** the vector of the random numbers.

**Author** Wenqiang Feng

**Email** von198@gmail.com

---

## CHAPTER SEVEN

---

# PUBLISHING ON GITHUB

---

### Chinese proverb

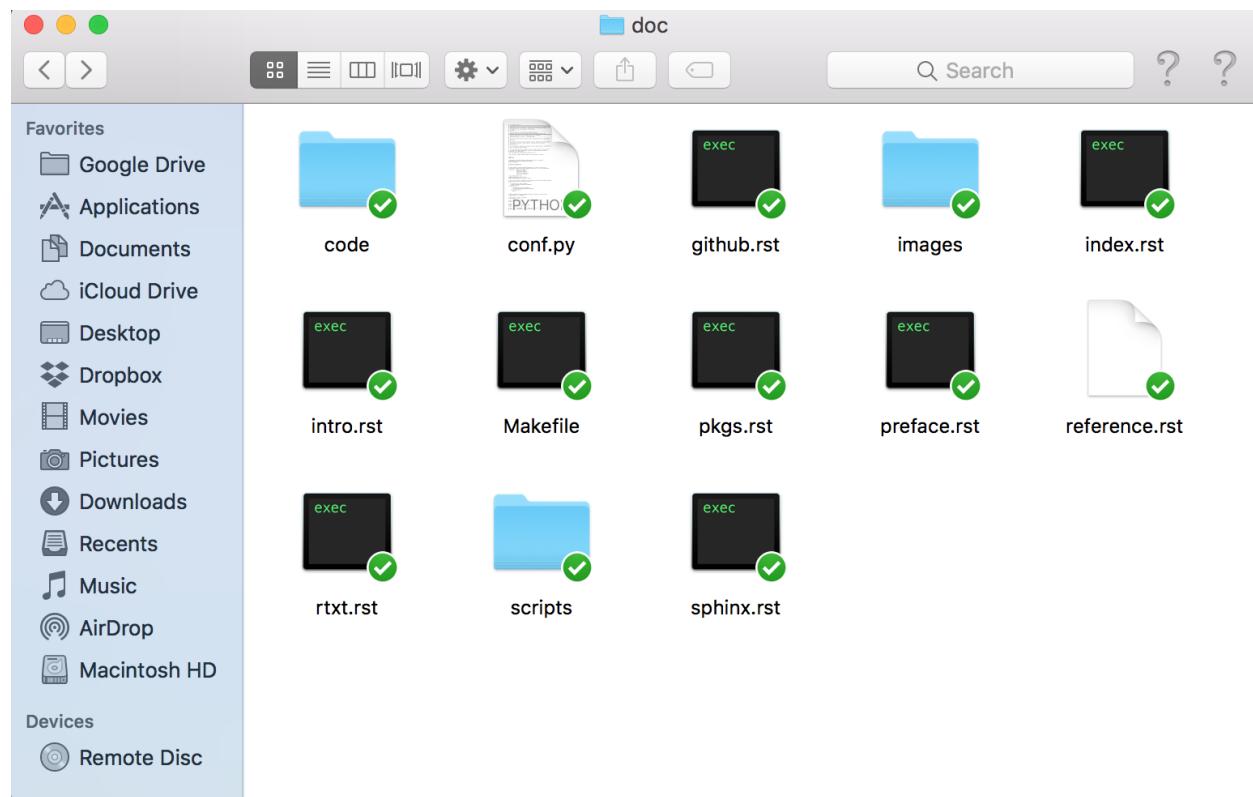
sharing your happiness is much better than enjoying your happiness by your own.

---

## 7.1 Create reStructuredTexts

### 7.1.1 Create reStructuredTexts

Use the tips in *reStructuredText Markup* to create your reStructuredTexts and add them in one folder, for example doc.



### 7.1.2 Add them in `index.rst`

```
=====
Contents
=====

.. toctree::
   :maxdepth: 2

preface
intro
pkgs
sphinx
rtxt
github
reference
```

### 7.1.3 Compile the reStructuredTexts files

1. Change the directory to the folder

```
cd MyTutorial/SphinxGithub/doc
```

2. Compile

```
make
```

Then you should get two more folders: `docs` and `latex`.

## 7.2 Create Repository on Github

## 7.3 Commit reStructuredTexts folder to Repository

Open Terminal and do the following steps:

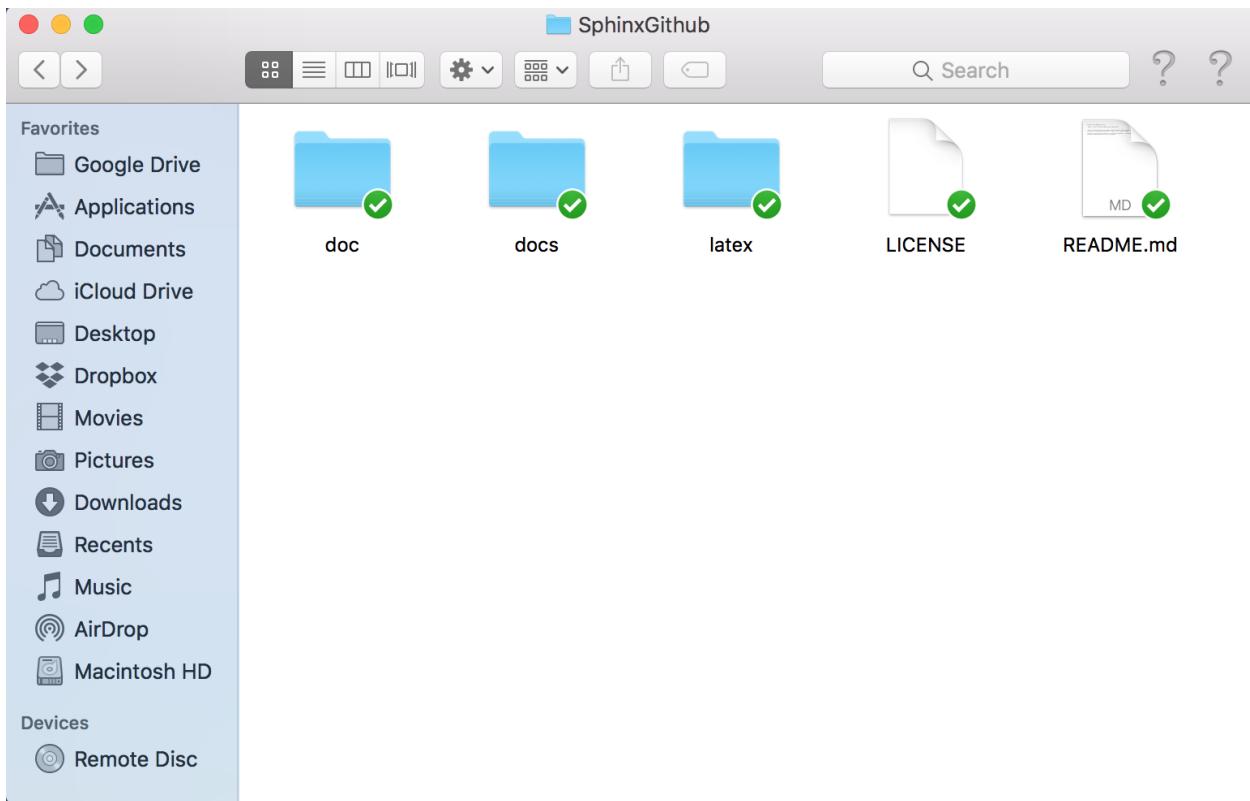
```
git init

git add .
# Adds the files in the local repository and stages them for commit. To unstage a file, use 'git reset HEAD YOUR-FILE'.

git commit -m "First commit"
# Commits the tracked changes and prepares them to be pushed to a remote repository. To remove this commit and modify the file, use 'git reset --soft HEAD~1' and commit and add the file again.

git remote add origin remote repository URL
```

(continues on next page)



(continued from previous page)

```
# Sets the new remote
git remote -v
# Verifies the new remote URL

git push -u origin master
# Pushes the changes in your local repository up to the remote repository you
→specified as the origin
```

## 7.4 Setup Github Pages on Github

### 7.4.1 Repository

Once you committed your repository to the Github, you will get:

### 7.4.2 Setup Github Pages

1. Go to Settings button in your repository
2. Set up the Github pages source
3. Enjoy your Github Pages

## Sphinx with Github Webpages, Release 1.00

---

### Create a new project

Coordinate, track, and update your work in one place, so projects stay transparent and on schedule.

#### Project board name

#### Description (optional)

#### Project template

Save yourself time with a pre-configured project board template.

#### Linked repositories

Search `runawayhorse001` to link repositories to this project for more accurate suggestions and better search results.

↪ Linked repositories: None yet!

**runawayhorse001 / SphinxGithub**

Code Issues Pull requests Projects Wiki Insights Settings

Sphinx Github Webpages Tutorial <https://runawayhorse001.github.io/Sph...> Edit

sph python3 latex pdf Manage topics

22 commits 1 branch 0 releases 1 environment 1 contributor MIT

Branch: master ▾ New pull request Create new file Upload files Find file Clone or download ▾

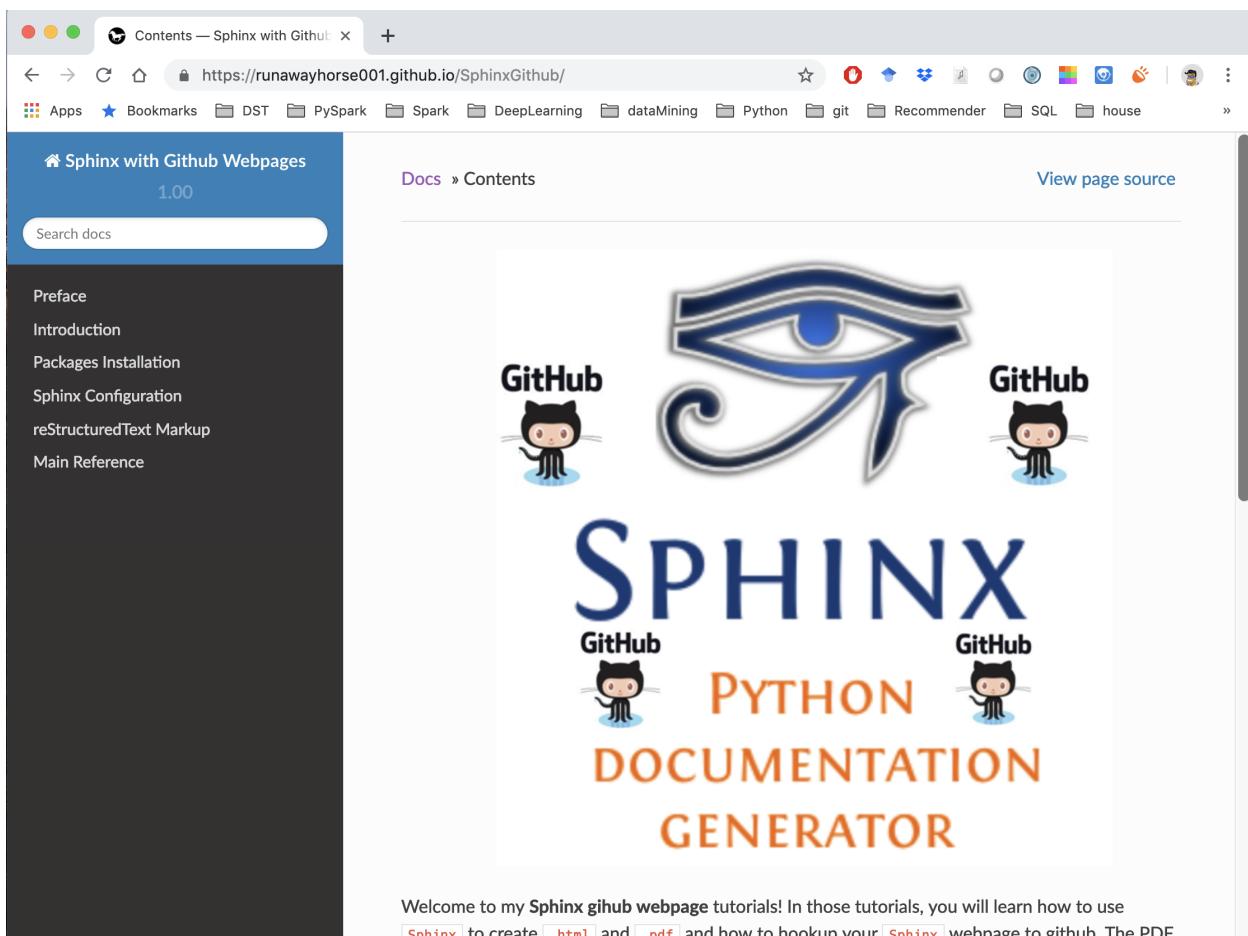
commit	message	time
runawayhorse001 and runawayhorse001 add icon	Latest commit a3c57a8 19 minutes ago	
doc	add icon	19 minutes ago
docs	add icon	19 minutes ago
latex	add icon	19 minutes ago
.gitignore	add .gitignore	2 days ago
LICENSE	Initial commit	3 days ago
README.md	add preface and introduction	2 days ago
README.md		

The screenshot shows the GitHub repository settings for 'SphinxGitHub'. The left sidebar contains options like Options, Collaborators, Branches, Webhooks, Notifications, Integrations & services, Deploy keys, Moderation, and Interaction limits. The main area has tabs for Settings, Repository name (set to 'SphinxGitHub'), and Features. Under Features, two checkboxes are checked: 'Wikis' (described as a simple way for others to contribute content) and 'Restrict editing to collaborators only' (described as making public wikis readable by everyone). At the top right, there are buttons for Unwatch (1), Unstar (1), Fork (0), and Settings.

## GitHub Pages

[GitHub Pages](#) is designed to host your personal, organization, or project pages from a GitHub repository.

The screenshot shows the GitHub Pages source configuration. It displays a green success message: '✓ Your site is published at <https://runawayhorse001.github.io/SphinxGitHub/>'. Below this, under 'Source', it says 'Your GitHub Pages site is currently being built from the /docs folder in the master branch.' with a 'Learn more' link. A dropdown menu shows 'master branch /docs folder' selected. A 'Save' button is next to it. A modal window titled 'Select source' lists three options: 'master branch' (disabled), 'master branch /docs folder' (selected and highlighted in blue), and 'None'. The 'master branch /docs folder' option is described as 'Use only the /docs folder for GitHub Pages.' A checked checkbox 'Enforce HTTPS' is present with the note '— Required for your site because you are using the default domain (runawayhorse001.github.io)'. At the bottom, a note states 'HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traffic to your site. When HTTPS is enforced, your site will only be served over HTTPS.' with a 'Learn more' link.



---

**CHAPTER  
EIGHT**

---

**MAIN REFERENCE**



# CHAPTER NINE

## APPENDIX: MY LATEX CHEAT SHEET

You can download the PDF version: [LaTeX Cheat Sheet](#).

Cheat Sheet for LATEX			
Wenqiang Feng E-mail: <a href="mailto:von198@gmail.com">von198@gmail.com</a> , Web: <a href="http://web.utk.edu/~wfeng1/">http://web.utk.edu/~wfeng1/</a>			
<b>1 LATEX Symbols</b>			
<b>1.1 package</b>			
\usepackage{amsymb, amsmath, amsthm, amsfonts}	\usepackage{multicol, multirow, calc, ifthen}	\usepackage{tikz, graphicx, color}	
\usepackage{algorithm, algpseudocode}	\usepackage[landscape]{geometry}		
<b>1.2 The basics</b>			
<b>description</b>	<b>command</b>	<b>output</b>	
plus or minus	\pm	$\pm$	
multiplication (times)	\times	$\times$	
multiplication (dot)	\cdot	$\cdot$	
division symbol	\div	$\div$	
backslash	\backslash	$\backslash$	
division (slash)	/	$/$	
circle plus	\oplus	$\oplus$	
circle times	\otimes	$\otimes$	
equal	\equiv	$\equiv$	
not equal	\neq	$\neq$	
less than or equal to	\leq	$\leq$	
greater than or equal to	\geq	$\geq$	
approximately equal to	\approx	$\approx$	
infinity	\infty	$\infty$	
fraction	\frac{a}{b}	$\frac{a}{b}$	
square root	\sqrt{x}	$\sqrt{x}$	
nth root	\sqrt[n]{x}	$\sqrt[n]{x}$	
exponentiation	a^b	$a^b$	
subscript	a_b	$a_b$	
absolute value	x	$ x $	
natural log	\ln(x)	$\ln(x)$	
logarithms	\log_a b	$\log_a b$	
exponential function	e^x	$e^x$	
pi	\pi	$\pi$	
degree	90^\circ	$90^\circ$	
dim	\dim(x)	$\dim(x)$	
det	\det(x)	$\det(x)$	
sin	\sin(x)	$\sin(x)$	
arcsin	\arcsin(x)	$\arcsin(x)$	
liminf	\liminf(x)	$\liminf(x)$	
dots	\ldots	$\ldots$	
dots	\cdots	$\cdots$	
diagonal dots	\ddots	$\ddots$	
underset	\overset{x}{\underset{y}{\overbrace{\dots}}}	$\overset{x}{\underset{y}{\overbrace{\dots}}}$	
overset	\overset{x}{\underset{y}{\overbrace{\dots}}}	$\overset{x}{\underset{y}{\overbrace{\dots}}}$	
<b>1.3 Define your own function</b>			
\newcommand{\norm}[1]{\left\  #1 \right\ }	\left\  #1 \right\	$\left\  #1 \right\ $	
\newcommand{\abs}[1]{\left  #1 \right }	\left  #1 \right	$\left  #1 \right $	
\newcommand{\lap}{\Delta}	\Delta	$\Delta$	
\newcommand{\tr}[1]{\operatorname{tr}(#1)}	\operatorname{tr}(#1)	$\operatorname{tr}(#1)$	
\newcommand{\tribint}[2]{\mathcal{E}^{#1}_{#2}}	\mathcal{E}^{#1}_{#2}	$\mathcal{E}^{#1}_{#2}$	
<b>1.4 Greek and Hebrew letters</b>			
<b>command</b>	<b>output</b>	<b>command</b>	<b>output</b>
\alpha	$\alpha$	\tau	$\tau$
\beta	$\beta$	\theta	$\theta$
\chi	$\chi$	\upsilon	$\upsilon$
\delta	$\delta$	\xi	$\xi$
\epsilon	$\epsilon$	\zeta	$\zeta$
\varepsilon	$\varepsilon$	\Delta	$\Delta$
\eta	$\eta$	\Gamma	$\Gamma$
\gamma	$\gamma$	\Lambda	$\Lambda$
\iota	$\iota$	\Omega	$\Omega$
\kappa	$\kappa$	\Phi	$\Phi$
\lambda	$\lambda$	\Pi	$\Pi$
\mu	$\mu$	\Psi	$\Psi$
\nu	$\nu$	\Sigma	$\Sigma$
\omega	$\omega$	\Theta	$\Theta$
\phi	$\phi$	\Upsilon	$\Upsilon$
\varphi	$\varphi$	\Xi	$\Xi$
\pi	$\pi$	\aleph	\aleph
<b>1.5 Geometry and trigonometry</b>			
<b>description</b>	<b>command</b>	<b>output</b>	
angle	\angle ABC	$\angle ABC$	
degree	90^\circ	$90^\circ$	
triangle	\triangle ABC	$\triangle ABC$	
segment	\overline{AB}	$\overline{AB}$	
sine	\sin(x)	$\sin(x)$	
cosine	\cos(x)	$\cos(x)$	
tangent	\tan(x)	$\tan(x)$	
cotangent	\cot(x)	$\cot(x)$	
secant	\sec(x)	$\sec(x)$	
cosecant	\csc(x)	$\csc(x)$	
inverse sine	\arcsin(x)	$\arcsin(x)$	
inverse cosine	\arccos(x)	$\arccos(x)$	
inverse tangent	\arctan(x)	$\arctan(x)$	
<b>1.6 Calculus</b>			
<b>description</b>	<b>command</b>	<b>output</b>	
derivative	\frac{df}{dx}	$\frac{df}{dx}$	
derivative	\frac{\partial f}{\partial x}	$\frac{\partial f}{\partial x}$	
partial derivative	\frac{\partial f}{\partial x}	$\frac{\partial f}{\partial x}$	
integral	\int	$\int$	
double integral	\iint	$\iint$	
triple integral	\iiint	$\iiint$	
limits	\lim_{x \rightarrow \infty}	$\lim_{x \rightarrow \infty}$	
summation	\sum_{n=1}^{\infty}	$\sum_{n=1}^{\infty}$	
<b>1.7 Linear algebra</b>			
<b>description</b>	<b>command</b>	<b>output</b>	
vector	\vec{v}	$\vec{v}$	
vector	\mathbf{v}	$\mathbf{v}$	
norm	\ v\	$\ v\ $	
matrix	\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix}	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix}$	
matrix	\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{pmatrix}	$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{pmatrix}$	
border matrix	\bordermatrix{& x & y & z\\ A & 1 & 2 & 3\\ B & 4 & 5 & 6\\ C & 7 & 8 & 0}	$\bordermatrix{& x & y & z\\ A & 1 & 2 & 3\\ B & 4 & 5 & 6\\ C & 7 & 8 & 0}$	
determinant	\begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{vmatrix}	$\begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{vmatrix}$	
determinant	\det(A)	$\det(A)$	
trace	\mathrm{tr}(A)	$\mathrm{tr}(A)$	
inverse	\mathrm{inv}(A)	$A^{-1}$	
transpose	\mathrm{trsp}(A)	$A^T$	
dimension	\dim(V)	$\dim(V)$	

©All Rights Reserved by Dr.Wenqiang Feng. Powered by LATEX. Updated 02-26-2015. von198@gmail.com

## 1.8 Logic

description	command	output
not	\sim	$\wedge$
and	\land	$\wedge$
or	\lor	$\vee$
if...then	\to	$\rightarrow$
if and only if	\Leftrightarrow	$\leftrightarrow$
logical equivalence	\equiv	$\equiv$
therefore	\therefore	$\therefore$
there exists	\exists	$\exists$
for all	\forall	$\forall$
implies	\Rightarrow	$\Rightarrow$
equivalent	\Leftrightarrow	$\leftrightarrow$

## 1.9 Set theory

description	command	output
element of	\in	$\in$
not an element of	\notin	$\notin$
subset of	\subset	$\subset$
subset of	\subseteq	$\subseteq$
not a subset of	\not\subseteq	$\not\subseteq$
contains	\supset	$\supset$
contains	\supseteq	$\supseteq$
union	\cup	$\cup$
intersection	\cap	$\cap$
big union	\bigcup_{n=1}^{10} A_n	$\bigcup_{n=1}^{10} A_n$
big intersection	\bigcap_{n=1}^{10} A_n	$\bigcap_{n=1}^{10} A_n$
empty set	\emptyset	$\emptyset$
power set	\mathcal{P}	$\mathcal{P}$

## 2 Figure

### 2.1 Insert figure

```
\begin{figure}[H]
\begin{center}
\includegraphics[width=0.2\textwidth]{fig.eps}
\caption{simulation result}\label{fig:pro2}
\end{center}
\end{figure}
```

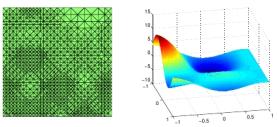


Figure 1: simulation result

## 2.2 draw figure with TikZ

```
\begin{figure}[H]
\begin{tikzpicture}[scale=0.5,>=latex]
\shade[ball color=gray,opacity=0.5] (0,0) circle (2cm);
\draw[->] (0,0,0) -- (3,0,0) node [xaxis] [right] {\$x\$};
\draw[->] (0,0,0) -- (0,3,0) node [yaxis] [right] {\$y\$};
\draw[->] (0,0,0) -- (0,0,3) node [zaxis] [left] {\$z\$};
\draw [->] (0,0,0)-- (2,0,2)node[right] {\$_{(r\sin\phi)}\$};
\draw [blue,->] (0,0,0)-- (0,0,5) node [yaxis] [left] {\$z\$};
\draw [blue,->] (0,0,0)-- (2,2,2)node[right] {\$_{(r\sin\phi)}\$};
\draw [blue,densely dashed] (2,0,2)--(2,1.8,0);
\draw [blue,densely dashed] (2,0,2)--(1.7,0,0)node[above] {\$x\$};
\draw [blue,densely dashed] (2,0,2)--(-0,0,1.9)node[left] {\$y\$};
\draw [blue,densely dashed] (2,0,2)--(0,0,0)node[below] {\$z\$};
\draw [blue,->] (0,0,0.5)to[bend right](0,0,5.0);
\node[] at (0.3,0.6,0) {\$_\phi\$};
\draw [blue,->] (0.5,0.5,0.5)to[bend right](0.5,0.5,0);
\node[] at (0.3,0.6,0) {\$_\phi\$};
\end{tikzpicture}
\caption{Spherical coordinate in \$\mathbb{R}^3\$}
\end{center}
\end{figure}
```

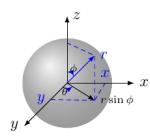


Figure 2: Spherical coordinate in  $\mathbb{R}^3$

## 3 Table

```
\begin{table}[H]
\caption{Errors of FEM for Poisson Equation}\label{tab:p2}
\begin{center}
\begin{tabular}{ccc}
\hline
h &  $\|u - u_h\|_{L^2}$  &  $|u - u_h|_{H^1}$  \\
\hline
1/4 &  $7.62 \times 10^{-3}$  &  $1.02 \times 10^{-1}$  \\
1/8 &  $2.02 \times 10^{-3}$  &  $5.58 \times 10^{-2}$  \\
1/16 &  $5.32 \times 10^{-4}$  &  $3.01 \times 10^{-2}$  \\
1/32 &  $1.37 \times 10^{-4}$  &  $1.60 \times 10^{-2}$  \\
1/64 &  $3.52 \times 10^{-5}$  &  $8.51 \times 10^{-3}$  \\
1/128 &  $8.87 \times 10^{-6}$  &  $4.48 \times 10^{-3}$  \\
1/256 &  $2.20 \times 10^{-6}$  &  $2.35 \times 10^{-3}$  \\
\hline
\end{tabular}
\end{center}

```

Table 1: Errors of FEM for Poisson Equation

## 4 Algorithm

```
\begin{algorithm}[H]
\caption{Bisection method}
\begin{algorithmic}[1]
\State \$a_0\$ gets a, \$b_0\$ gets b\$\\
\While{\$k > \$}
\State \$c_k\$ gets \frac{a_{(k-1)}+b_{(k-1)}}{2}\$\\
\If {\$f(a_{(k)})f(c_{(k)}) < 0\$}
\State \$a_k\$ gets a_{(k-1)}\$\\
\State \$b_k\$ gets c_{(k)}\$\\
\EndIf\\
\If {\$f(b_{(k)})f(c_{(k)}) < 0\$}
\State \$a_k\$ gets c_{(k)}\$\\
\State \$b_k\$ gets b_{(k-1)}\$\\
\EndIf\\
\State \$c_k\$ gets c_{(k)}gets \frac{a_{(k)}+b_{(k)}}{2}\$\\
\EndWhile
\end{algorithmic}
\end{algorithm}
```

### Algorithm 1 Bisection method

```

1:  $a_0 \leftarrow a, b_0 \leftarrow b$ 
2: while  $k < 0$  do
3:    $c_k \leftarrow \frac{a_{k-1} + b_{k-1}}{2}$ 
4:   if  $f(a_k)f(c_k) < 0$  then
5:      $a_k \leftarrow a_{k-1}$ 
6:      $b_k \leftarrow c_k$ 
7:   end if
8:   if  $f(b_k)f(c_k) < 0$  then
9:      $a_k \leftarrow c_k$ 
10:     $b_k \leftarrow b_{k-1}$ 
11:  end if
12:   $x^k \leftarrow c^k \leftarrow \frac{a_k + b_k}{2}$ 
13: end while

```

© All Rights Reserved by Dr. Wenqiang Feng. Powered by L<sup>A</sup>T<sub>E</sub>X. Updated 02-26-2015. vov198@gmail.com.

## BIBLIOGRAPHY

[Ref] Book or article reference, URL or whatever.

[Sphinx2019] Sphinx Team. [SPHINX: Python Documentation Generator](#).

[Georg2018] Georg Brandl. [Sphinx Documentation, Release 1.7.10+](#), 2018



## PYTHON MODULE INDEX

### S

`statspy.basics`, 55

`statspy.tests`, 56



## INDEX

### R

`rnorm()` (*in module statspy.basics*), 55

### S

`statspy.basics` (*module*), 55

`statspy.tests` (*module*), 56

### T

`t_test()` (*in module statspy.tests*), 56