

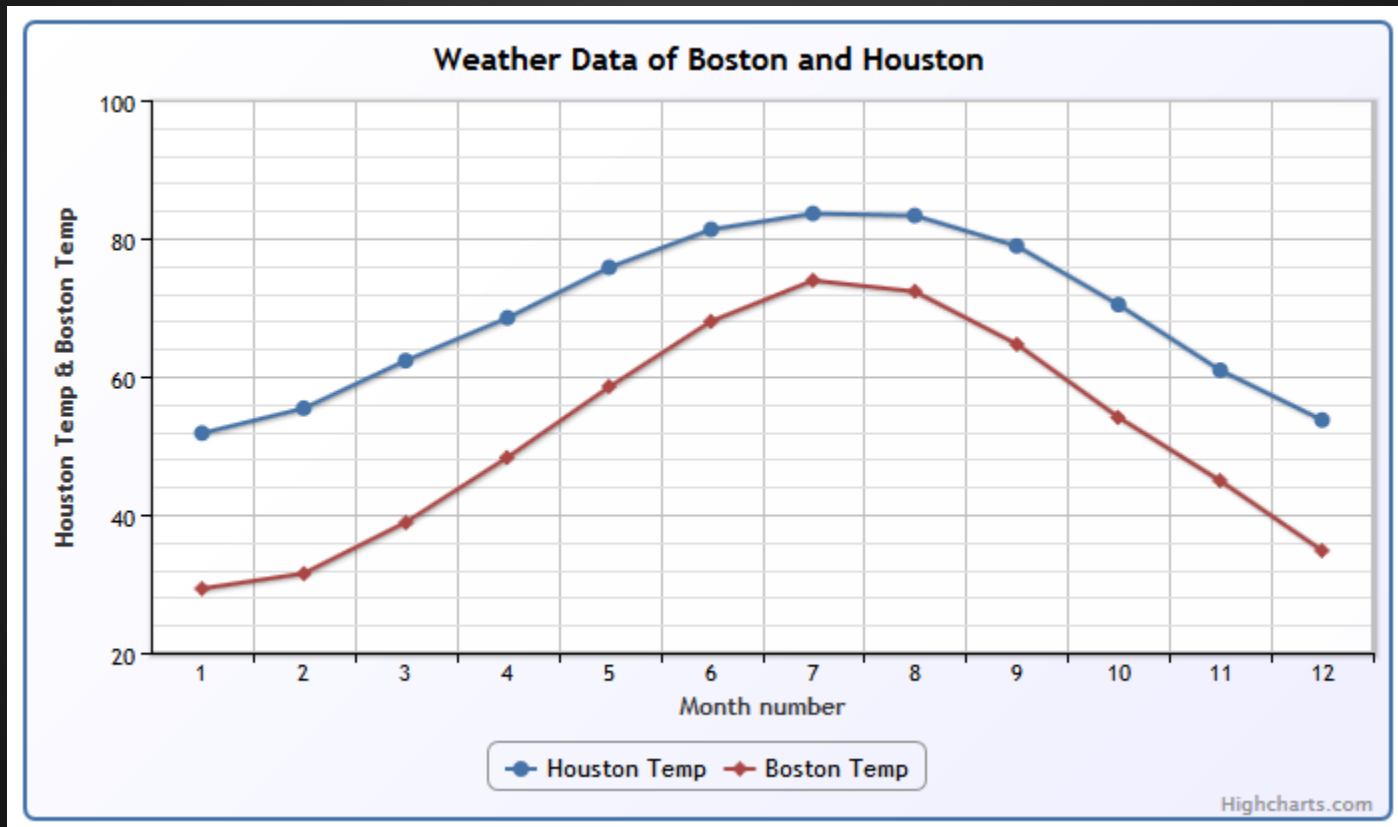
# Django + Charts = Good

3 options

# Django + Charts = Good

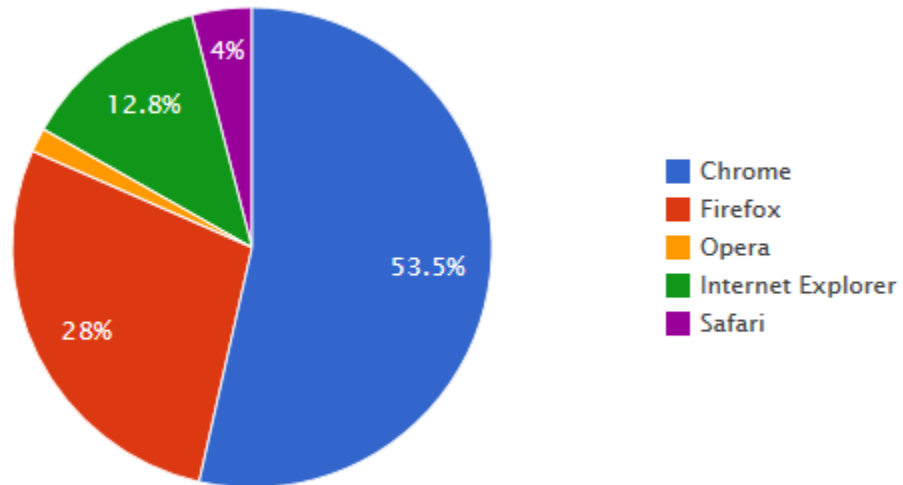
- What's Django ?
- What's Charts ?
- What's Good ?

# Like this ...



... and this

Browser Statistics, May 2013



## 3 Libraries

- Chartit
- Chartkick
- Django-Graphos

<http://chartit.shutupandship.com/>

<https://github.com/mher/chartkick.py>

<http://chartit.shutupandship.com/>

# What to do ?

- Simple chart implemented in each library
- How ?
- How hard for previous non-user ?
- Does it do what it says on the label ?

**Before we start**

**Join in the chorus ...**

<https://bitbucket.org/rshea/django-charts-demo>

A simple chart implemented in each of the  
candidate libraries



## Some Basics common to all

- Rely on Javascript library to draw chart
- ... but you don't need to know about it !
  
- Library specific stuff into Django template
- Library specific stuff into Django view
  
- I create an app for each: dmochrtit, dmochrkck, dmodjgrap

**And so to ChartIt**

# Chartit

- <http://chartit.shutupandship.com>
- Used HighCharts.js to draw chart in browser

The logo for 'django chartit' is displayed on a white rectangular background. The word 'django' is written in a bold, black, lowercase sans-serif font. The word 'chartit' is written in a bold, olive-green, lowercase sans-serif font, positioned to the right of 'django' with a small gap between them.

**django chartit**

# Chartit - 30 second how-to

- `pip install django-chartit`
- `pip install simplejson` (fixes a problem that occurs with some data types)
- Download HighCharts and jQuery to `static` directory of host application
- Add host application to `INSTALLED_APPS` in settings.
- Add `index` method to `views.py` in host application.
- Add template for use by `index` method to host application.

# Chartit - 3 minute how-to

settings.py

```
INSTALLED_APPS = (  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.sites',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'suit',  
    'chartit',  
    'graphos',  
    'chartkick',  
    'dcdemo.apps.dmodrv',  
    'dcdemo.apps.dmochrTit',  
    'dcdemo.apps.dmodjgrap',  
    'dcdemo.apps.dmochrkck',  
    'django.contrib.admin',  
    # Uncomment the next line to enable admin documentation:  
    # 'django.contrib.admindocs',  
)
```

# Chartit - 3 minute how-to

ChartIt provides a DataPool object to provide a data feed.

And a Chart object to actually render the chart

Like this :

# Chartit - 3 minute how-to

views.py

```
def chart1(request):
    #Step 1: Create a DataPool with the data we want to retrieve.
    weatherdata = \
        DataPool(
            series=
            [{'options': {
                'source': WindObservation.objects.all()},
                'terms': [
                    'when',
                    'speed']}]
        )

    #Step 2: Create the Chart object
    cht = Chart(
        datasource = weatherdata,
        series_options =
            [{'options':{
                'type': 'line',
```

# Chartit - 3 minute how-to

chart1.html

```
<h1>Django-chartit Chart 1</h1>
<head>
  <!-- code to include the highcharts and jquery libraries goes here -->
  <script src="/static/dmochrtit/js/jquery-1.9.1.js" type="text/javascript"
  <script src="/static/dmochrtit/js/highcharts/highcharts.js" type="text/ja
  <!-- load_charts filter takes a comma-separated list of id's where -->
  <!-- the charts need to be rendered to -->
  {% load chartit %}
  {{ weatherchart|load_charts:"container" }}
</head>
```

```
</div>
<div id="container">chart will be rendered here</div>
```



**And now the output !**

# Or not as the case maybe

- ChartIt struggles with DateTime
- ... my model had DateTime !
- Relevant (unresolved) issues:
  - <https://github.com/pgollakota/django-chartit/issues/8>
  - <https://github.com/pgollakota/django-chartit/issues/12>


That was **ChartIt** this is  
**ChartKick**

Confusing - huh ?

# Chartkick

- <https://github.com/mher/chartkick.py>
- Uses HighCharts.js or Google Chart to draw

## Chartkick

Create beautiful Javascript charts with one line of Ruby 

Also available in Python and pure JavaScript

# Chartkick - 30 second how-to

- `pip install chartkick`
- Download Flot to `static` directory of host application (see my sample code for what I left in from git download)
- Add 'chartkick' to `INSTALLED_APPS` in settings.
- Add host application to `INSTALLED_APPS` in settings.
- Add 'chartkick.js' to `STATICFILES_DIRS`
- Add suitable method, eg `chart1`, method to `views.py` in host application.
- Revise `urls.py` to route a url to the `views.py` method just created.
- Add template for use by `index` method to host application.

# Chartkick - 3 minute how-to

settings.py

```
import chartkick
PROJECT_DIR = os.path.dirname(__file__)
STATICFILES_DIRS = (os.path.join(PROJECT_DIR, 'static'),)
STATICFILES_DIRS = (
    chartkick.js(),
)
```

# Chartkick - 3 minute how-to

settings.py

```
INSTALLED_APPS = (  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.sites',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'suit',  
    'chartit',  
    'graphos',  
    'chartkick',  
    'dcdemo.apps.dmodrv',  
    'dcdemo.apps.dmochr tit',  
    'dcdemo.apps.dmodjgrap',  
    'dcdemo.apps.dmochrkck',  
    'django.contrib.admin',  
    # Uncomment the next line to enable admin documentation:  
    # 'django.contrib.admindocs',  
)
```

# Chartkick - 3 minute how-to

Chartkick also struggles with DateTime objects but we can overcome it more easily than with ChartIt

- Get your queryset
- Migrate into ISO date keyed dictionary
- Include dictionary in Django template context



# Charkick - 3 minute how-to

views.py

```
#Get some wind observations for a fixed period and location
utc = pytz.utc
from dcdemo.apps.dmodrv.models import WindObservation
wo = WindObservation.objects.filter(
    when__gte=datetime.datetime(2005, 1, 1, 0, 0, 0, tzinfo=utc),
    when__lte=datetime.datetime(2005, 1, 1, 23, 59, 59, tzinfo=utc),
    station_id=1)

#Build a dictionary of string value keyed by ISO date/times
exchange = {}
for windobs in wo:
    exchange[windobs.when.strftime('%Y%m%dT%H%M%S')] = str(windobs.speed)

template = loader.get_template('dmochrkck/chart1.html')
context = RequestContext(request, {})
context["exchange"] = exchange
return HttpResponse(template.render(context))
```

# Charkick - 3 minute how-to

views.py

```
#Get some wind observations for a fixed period and location
utc = pytz.utc
from dcdemo.apps.dmodrv.models import WindObservation
wo = WindObservation.objects.filter(
    when__gte=datetime.datetime(2005, 1, 1, 0, 0, 0, tzinfo=utc),
    when__lte=datetime.datetime(2005, 1, 1, 23, 59, 59, tzinfo=utc),
    station_id=1)

#Build a dictionary of string value keyed by ISO date/times
exchange = {}
for windobs in wo:
    exchange[windobs.when.strftime('%Y%m%dT%H%M%S')] = str(windobs.speed)

template = loader.get_template('dmochrkck/chart1.html')
context = RequestContext(request, {})
context["exchange"] = exchange
return HttpResponse(template.render(context))
```

# Charkick - 3 minute how-to

views.py

```
#Get some wind observations for a fixed period and location
utc = pytz.utc
from dcdemo.apps.dmodrv.models import WindObservation
wo = WindObservation.objects.filter(
    when__gte=datetime.datetime(2005, 1, 1, 0, 0, 0, tzinfo=utc),
    when__lte=datetime.datetime(2005, 1, 1, 23, 59, 59, tzinfo=utc),
    station_id=1)

#Build a dictionary of string value keyed by ISO date/times
exchange = {}
for windobs in wo:
    exchange[windobs.when.strftime('%Y%m%dT%H%M%S')] = str(windobs.speed)

template = loader.get_template('dmochrkck/chart1.html')
context = RequestContext(request, {})
context["exchange"] = exchange
return HttpResponse(template.render(context))
```

# Chartkick - 3 minute how-to

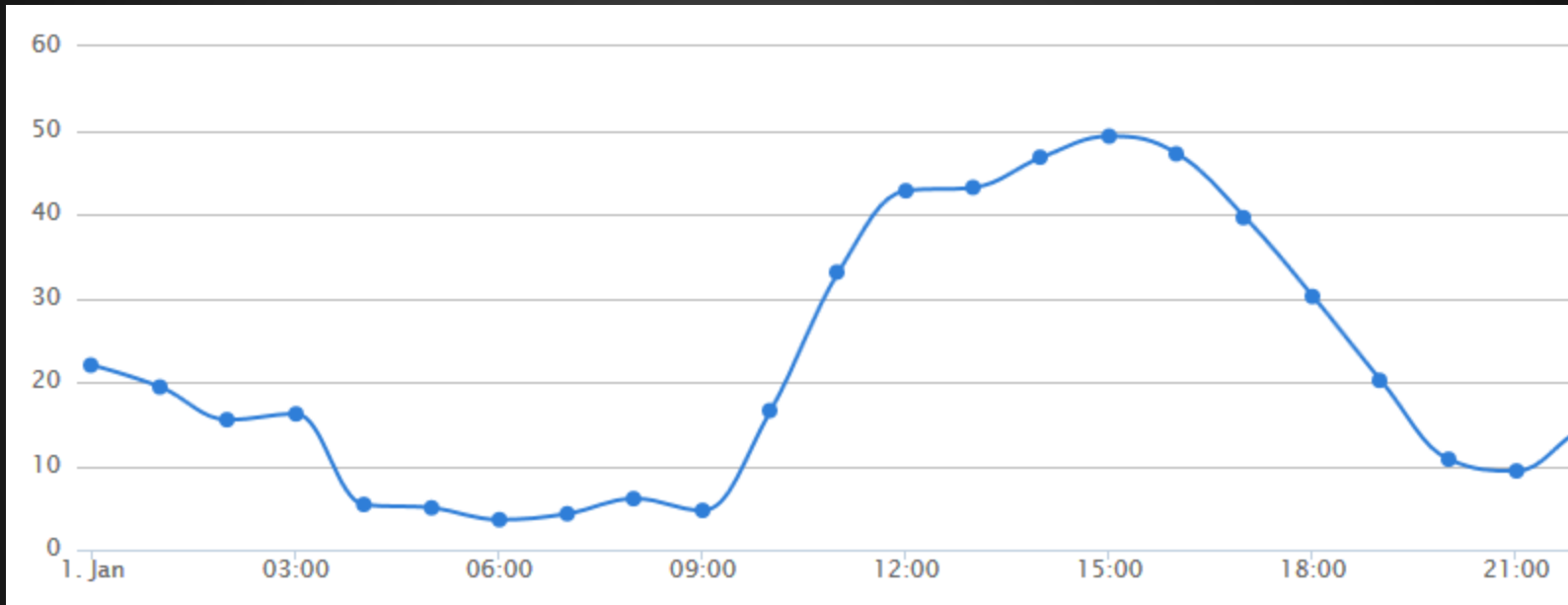
chart1.html

```
{% load chartkick %}
<h1>chartkick Chart 1</h1>
<head>
  <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
  <script src="http://code.highcharts.com/highcharts.js"></script>
  <script src="/static/dmochrkck/js/chartkick.js"></script>
</head>
```

```
<div id="container">
  {% line_chart exchange with min=0.0 id='rates' %}
</div>
```

**And now the output !**

# Chartkick output



# Chartkick output

- Great looking chart (thanks to HighLight.js)
- Tooltips on data points
- How to label Axis from Chartkick ?
- How to label chart from Chartkick ?

That was **ChartKick** this is  
django-graphos

Last one - I promise



# django-graphos

- <https://github.com/agiliq/django-graphos>

Django Graphos 0.0.2a0 documentation »

## Table Of Contents

Welcome to Django Graphos's  
documentation!  
Indices and tables

# Welcome to Django Graphos

Contents:

# django-graphos

- <https://github.com/agiliq/django-graphos>
- A lot of different drawing options:
  - Flot
  - Google Charts API
  - YUI Charts
  - Morris.js
  - Highcharts
- I'm using Flot ([flotcharts.org](http://flotcharts.org))

# django-graphos - 30 second how-to

- `pip install django-graphos`
- Download Flot to `static` directory of host application (see my sample code for what I left in from git download)
- Add host application to `INSTALLED_APPS` in settings.
- Add 'graphos' to `INSTALLED_APPS` in settings.
- Add suitable method, eg `chart1`, method to `views.py` in host application.
- Revise `urls.py` to route a url to the `views.py` method just created.
- Add template for use by `index` method to host application.

# django-graphos - 3 minute how-to

settings.py

```
INSTALLED_APPS = (  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.sites',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'suit',  
    'chartit',  
    'graphos',  
    'chartkick',  
    'dcdemo.apps.dmodrv',  
    'dcdemo.apps.dmochr tit',  
    'dcdemo.apps.dmodjgrap',  
    'dcdemo.apps.dmochrkek',  
    'django.contrib.admin',  
    # Uncomment the next line to enable admin documentation:  
    # 'django.contrib.admindocs',  
)
```

# django-graphos - 3 minute how-to

Guess what ? django-graphos struggles with DateTime's in your Django models

- Get your queryset
- Save X/Y's into list of lists
- Include list of lists in Django template context

# django-graphos - 3 minute how-to

views.py

```
#Get some wind observations for a fixed period and location
utc = pytz.utc
from dcdemo.apps.dmodrv.models import WindObservation
wo = WindObservation.objects.filter(
    when__gte=datetime.datetime(2005, 1, 1, 0, 0, 0, tzinfo=utc),
    when__lte=datetime.datetime(2005, 1, 1, 23, 59, 59, tzinfo=utc),
    station_id=1)

#Build a list of lists; each inner list being a x/y data point
chart_data = []
chart_data.append(['Hour (24H)', 'Speed (Km/h)'])
for windobs in wo:
    chart_data.append([windobs.when.hour, str(windobs.speed)])
```

# django-graphos - 3 minute how-to

views.py

```
#Get some wind observations for a fixed period and location
utc = pytz.utc
from dcdemo.apps.dmodrv.models import WindObservation
wo = WindObservation.objects.filter(
    when__gte=datetime.datetime(2005, 1, 1, 0, 0, 0, tzinfo=utc),
    when__lte=datetime.datetime(2005, 1, 1, 23, 59, 59, tzinfo=utc),
    station_id=1)
```

```
#Build a list of lists; each inner list being a x/y data point
chart_data = []
chart_data.append(['Hour (24H)', 'Speed (Km/h)'])
for windobs in wo:
    chart_data.append([windobs.when.hour, str(windobs.speed)])
```

# django-graphos - 3 minute how-to

views.py

```
#The following dictionary illustrates how we can directly
#provide options to the flot chart which correspond to the
#normal flot options. We provide a python dict which corresponds
#to the javascript object which would normally be provided
#
#In this case we're ensuring that we don't end up with axis
#label values which are something other than integers
opt = {
    'xaxis': {
        'tickDecimals': 0,
    },
    'yaxis': {
        'tickDecimals': 0,
    }
}
```



# django-graphos - 3 minute how-to

Because django-graphos is using Flot we can use the 'Flot Chart Options' structure.

<https://github.com/flot/flot/blob/master/API.md#plot-options>

## Plot Options

All options are completely optional. They are documented

```
var options = {
  series: {
    lines: { show: true },
    points: { show: true }
  }
};

$.plot(placeholder, data, options);
```

# django-graphos - 3 minute how-to

views.py

```
line_chart = flot.LineChart(SimpleDataSource(data=chart_data), html_id="line_chart", options=opt)
context = RequestContext(request, {})
context["line_chart"] = line_chart
template = loader.get_template('dmodjgrap/chart1.html')
return HttpResponse(template.render(context))
```

# django-graphos - 3 minute how-to

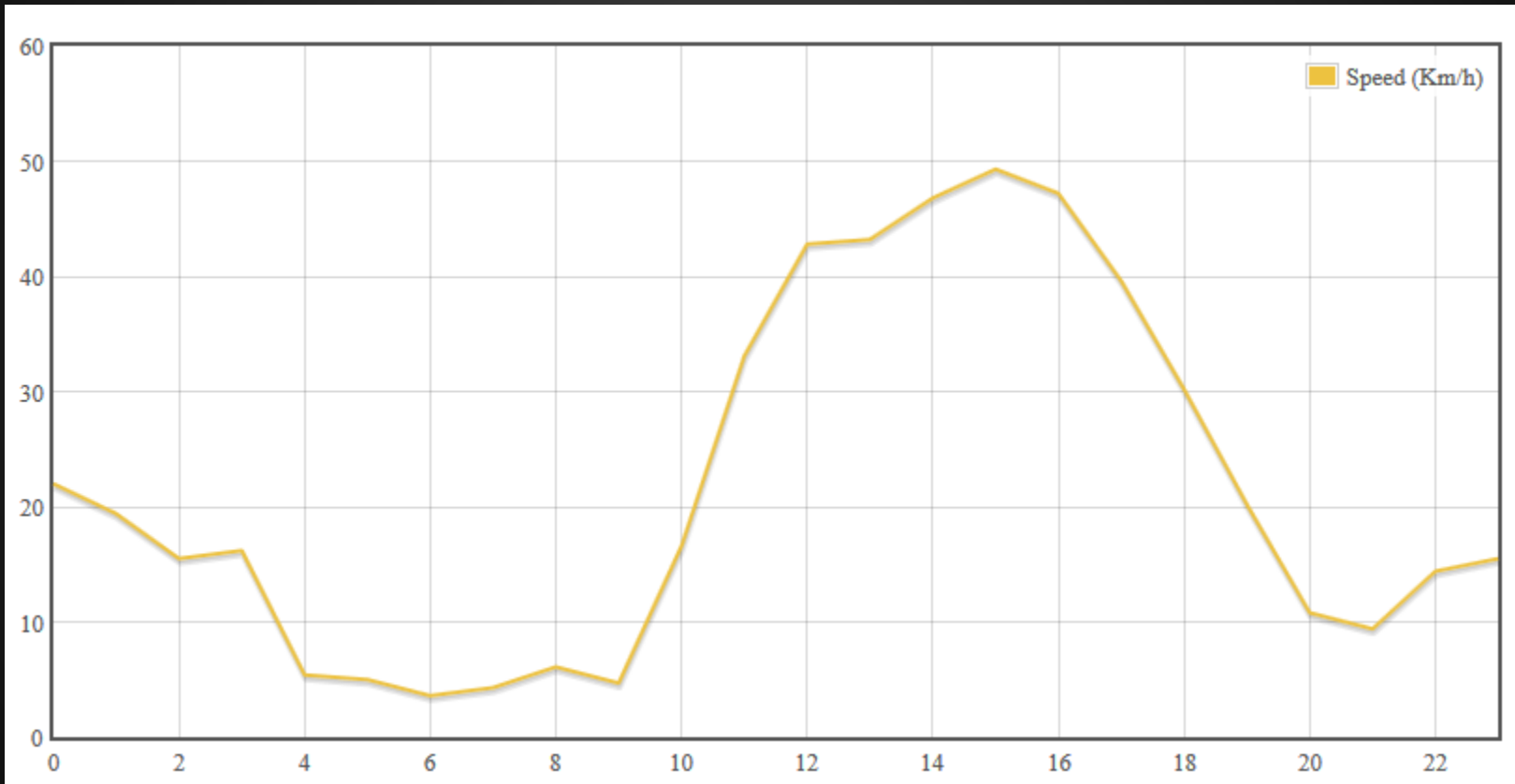
chart1.html

```
<script type="text/javascript" src="/static/dmodjgrap/js/flot/jquery.js"></script>
<!--[if lte IE 8]><script language="javascript" type="text/javascript" src="/static
<script src="/static/dmodjgrap/js/flot/jquery.flot.js"></script>
<script src="/static/dmodjgrap/js/flot/jquery.flot.time.js"></script>
```

```
<div id="container">
    {{ line_chart.as_html }}
</div>
```

**And now the output !**

# django-graphos output



# django-graphos output

- More than good looking enough
- Drop-in to options very powerful
- Attempts to label chart failed

# Conclusions

# Chartit - conclusions

- Not very pythonic ?
- Limited variety of charts
- DateTime defects - FAIL !
- Not my favourite



# Chartkick - conclusions

- Good if you want HighCharts/Google Chart
- DateTime - need for workaround  
disappointing
- OK but comes in second

# Django-graphos conclusions

- Wide variety of outputs
- DateTime - the strangeness continues
  - Actually more optimistic I could make it work directly
- Developers were friendly when asked for help
- My number 1

