



# Go for Data Science

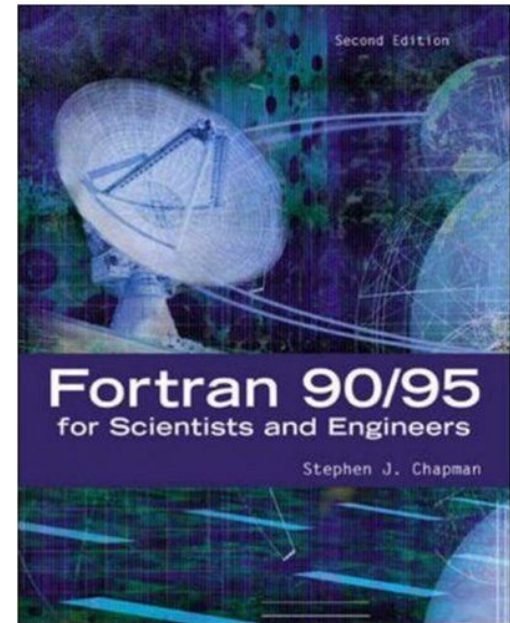
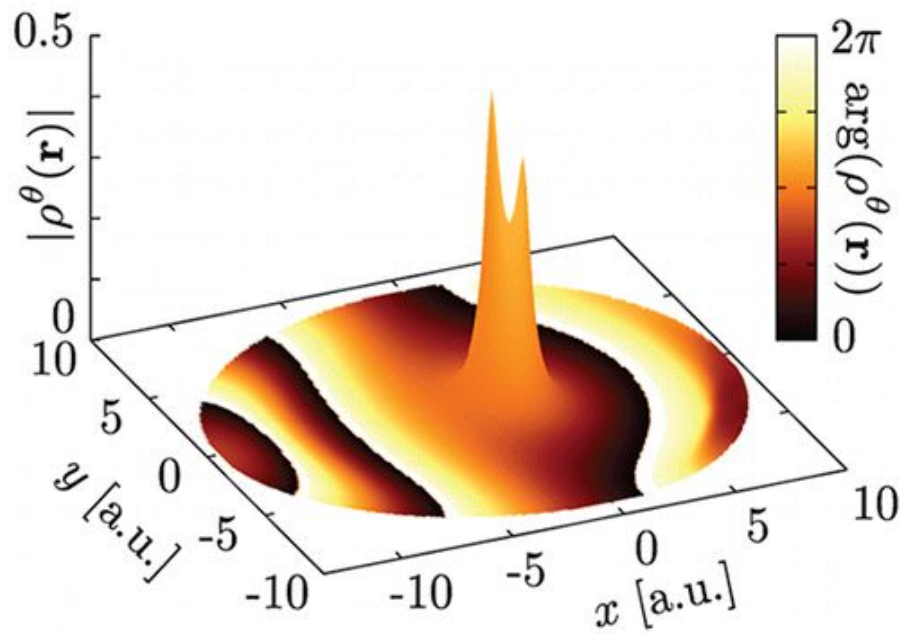
Daniel Whitenack, Ardan Labs

@dwhitena, #Gophercon, #Godatascience

# Outline

- Intro
- What is data science?
- Data science struggles.
- Data science with Go.
- Get started, contribute.

# Intro



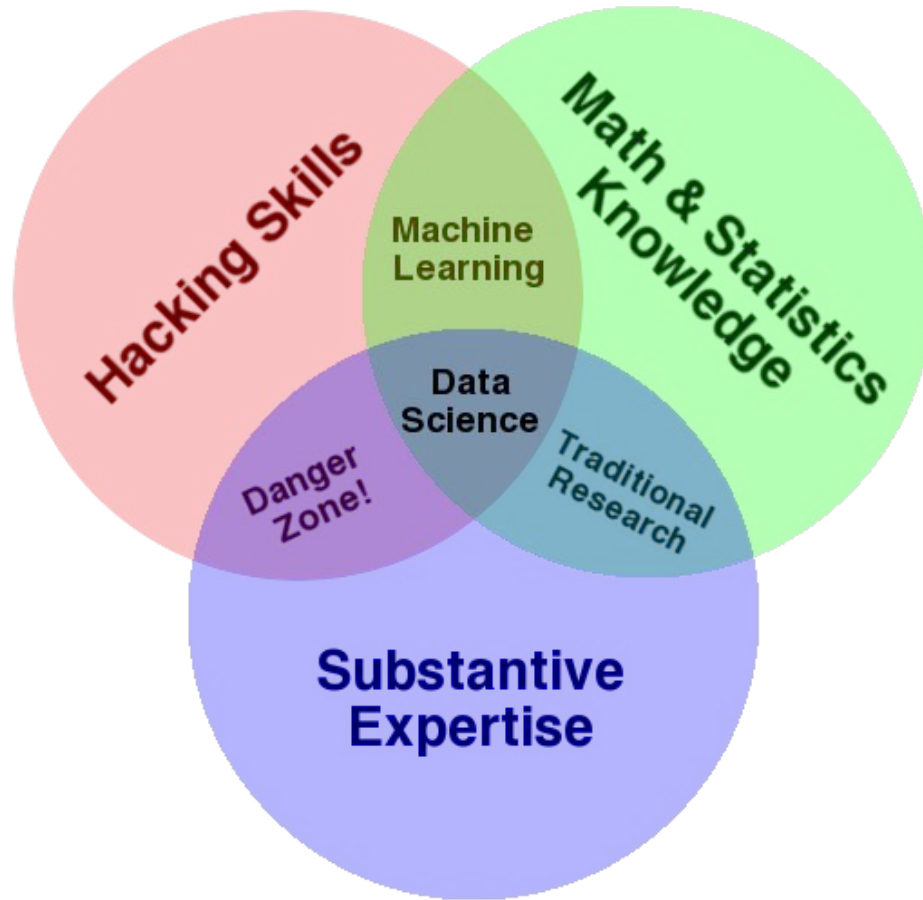


Image from [Forbes](#)



Telnyx



Python

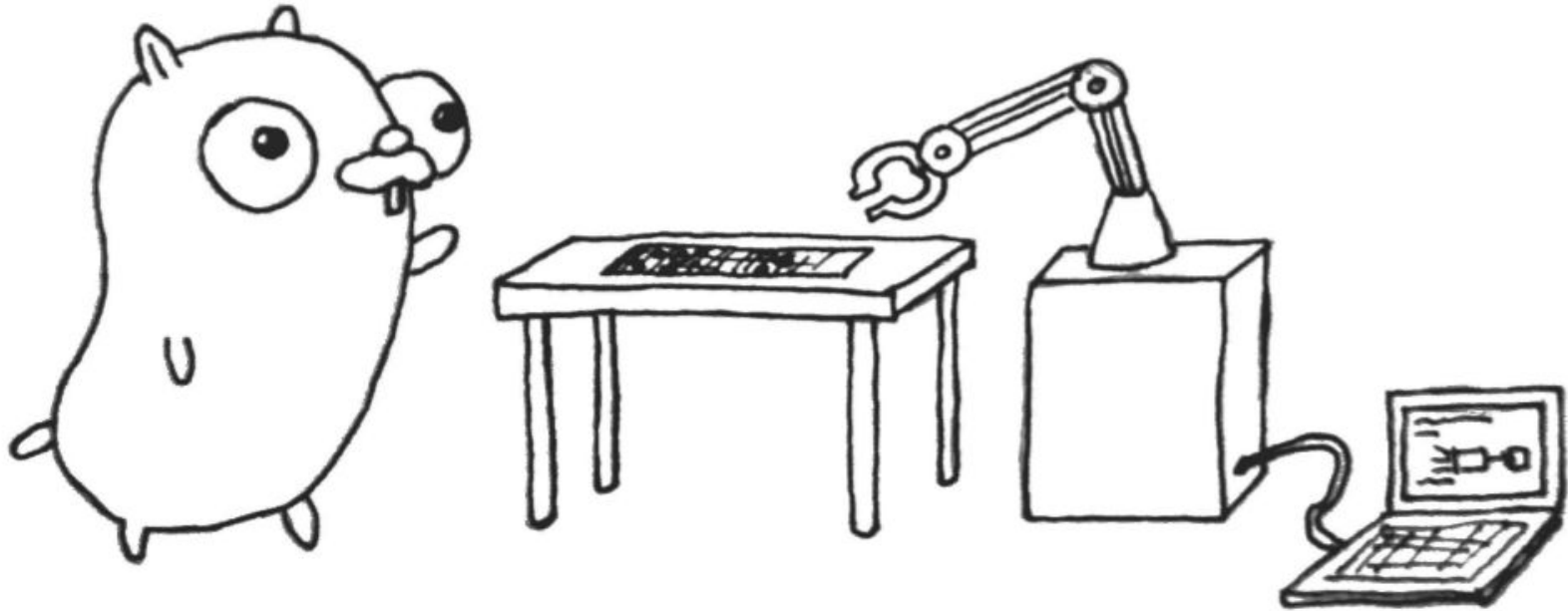


Go

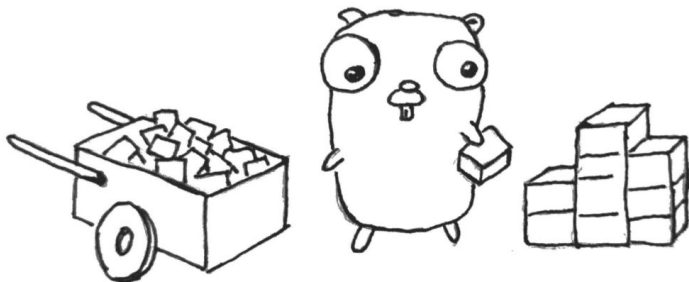
# What is data science?



# #datascience



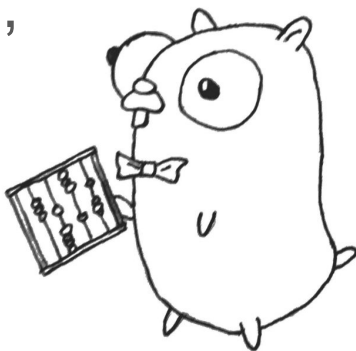
# Data Science



ETL, Data Cleaning,  
Organization



Parsing, Extraction of  
Patterns



Arithmetic

# Data science struggles

# Integrity and Reproducibility

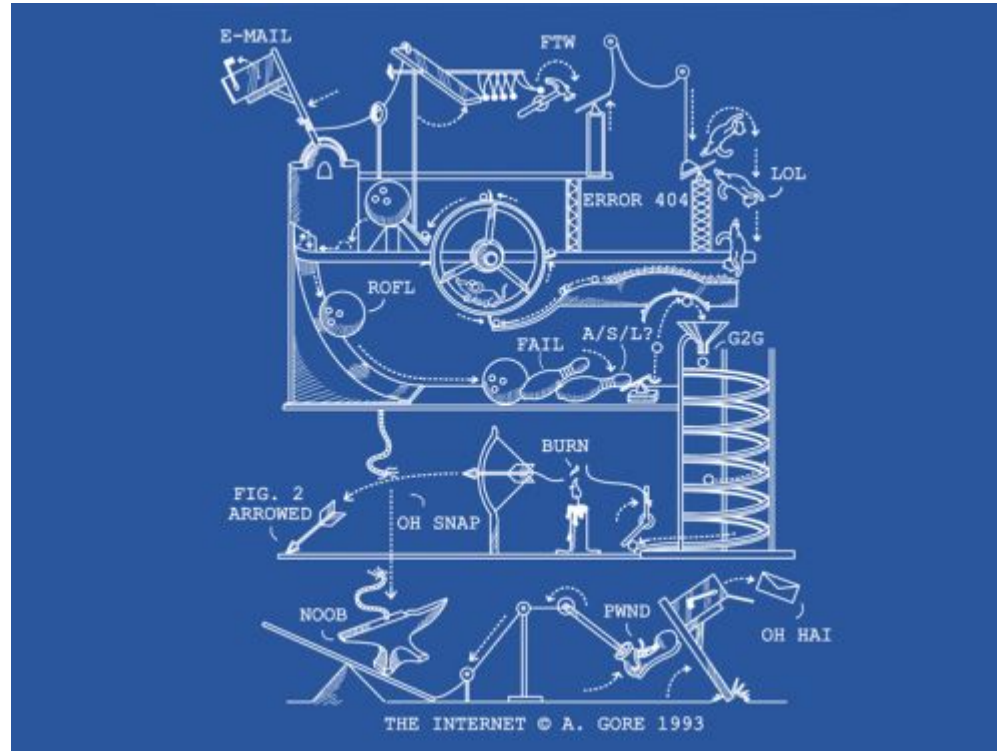


Image from [here](#)

# Integrity and Reproducibility



```
$ cat example.csv  
1,testval1  
2,testval2  
3,testval3
```

```
import pandas as pd
```

```
cols = [  
    'integercolumn',  
    'stringcolumn'  
]
```

```
data = pd.read_csv('example.csv',  
                   names=cols)
```

```
print data['integercolumn'].max()
```



```
f, err := os.Open("example.csv")
if err != nil {
    err = errors.Wrap(err, "Could not open CSV")
    log.Fatal(err)
}

r := csv.NewReader(bufio.NewReader(f))
records, err := r.ReadAll()
if err != nil {
    err = errors.Wrap(err, "Could not parse CSV")
    log.Fatal(err)
}
```





```
var intMax int
for _, record := range records {
    intVal, err := strconv.Atoi(record[0])
    if err != nil {
        err = errors.Wrap(err, "Parse failed")
        log.Fatal(err)
    }
    if intVal > intMax {
        intMax = intVal
    }
}
fmt.Println(intMax)
```





```
$ python example.py  
3
```



```
$ go run example.go  
3
```

```
$ cat example.csv  
1,testval1  
2,testval2  
,testval3
```



```
$ python example2.py  
2.0
```



```
$ go run example2.go  
2016/05/12 13:23:53 Parse  
failed: strconv.ParseInt:  
parsing "": invalid  
syntax  
exit status 1
```

# Integrity and Deployment



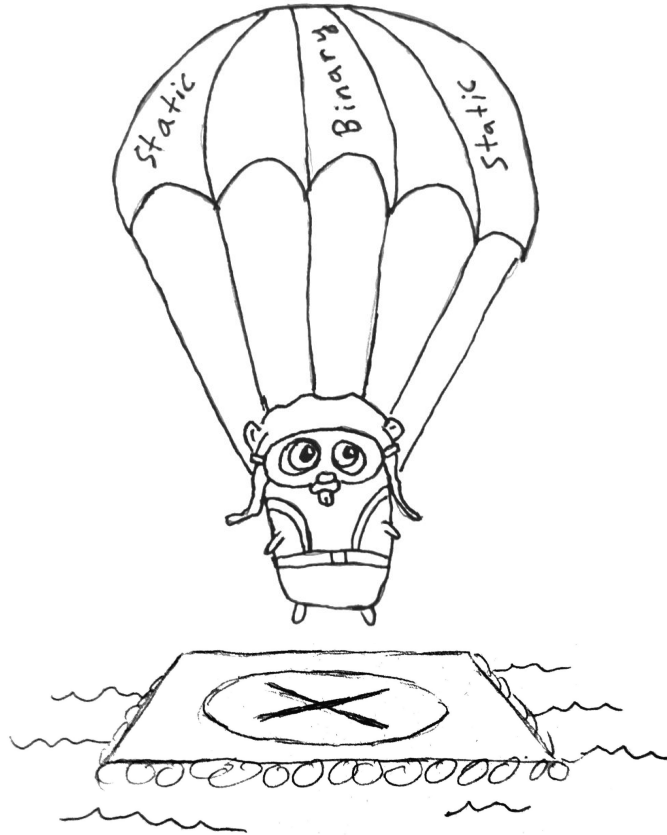
```

FROM ubuntu:latest
ENV DEBIAN_FRONTEND noninteractive
ENV PATH /anaconda/bin:$PATH
# For image inheritance.
ONBUILD ENV PATH /anaconda/bin:$PATH
# Install packages ... change the timezone line if you're not in Pacific
time
RUN apt-get -y update && apt-get install -y wget nano locales curl unzip
wget openssl libhdf5-dev libpq-dev \
    && apt-get clean && dpkg-reconfigure locales && locale-gen en_US.UTF-8
\
    && echo "America/Los_Angeles" > /etc/timezone && dpkg-reconfigure --
frontend noninteractive tzdata \
    && apt-get autoremove \
    && rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*
# Install and setup minimal Anaconda Python distribution, then tear down
temp files
RUN wget http://repo.continuum.io/miniconda/Miniconda-latest-Linux-x86_64.
sh -O miniconda.sh \
    && bash miniconda.sh -b -p /anaconda \
    && conda install scipy numpy scikit-learn nose readline pandas
matplotlib seaborn dateutil ipython-notebook nltk \
    pip psycpg2 cython hdf5 pytables ipywidgets \
    && conda clean -i -l -t -y \
    && rm miniconda.sh
# Get the pip packages and clean up
ADD requirements.txt /
RUN pip install -r /requirements.txt && rm -rf /root/.cache/pip/*
ENV LANGUAGE en_US.UTF-8
ENV LANG en_US.UTF-8
ENV PEM_FILE /key.pem
# $PASSWORD will get `unset` within notebook.sh, turned into an IPython
style hash
ENV PASSWORD Dont make this your default
ENV USE_HTTP 1
# Add current files to / and set entry point.
ADD . /workspace
WORKDIR /workspace
ADD notebook.sh /notebook.sh
RUN chmod a+x /notebook.sh
EXPOSE 8888
CMD ["/notebook.sh"]

```



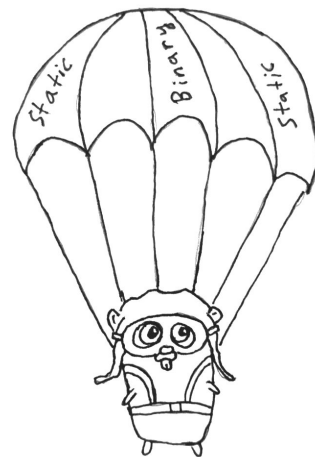
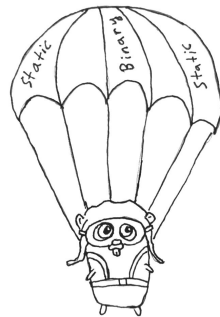
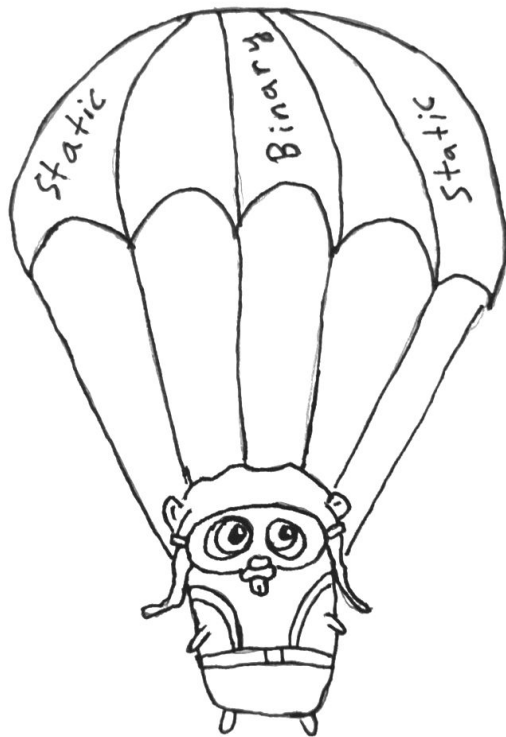
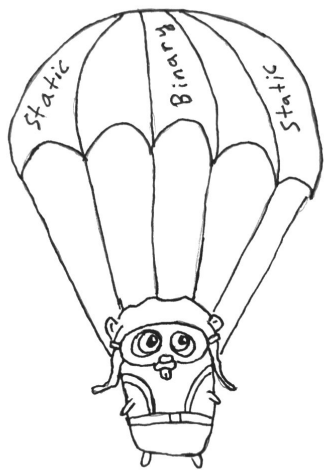
# Integrity and Deployment





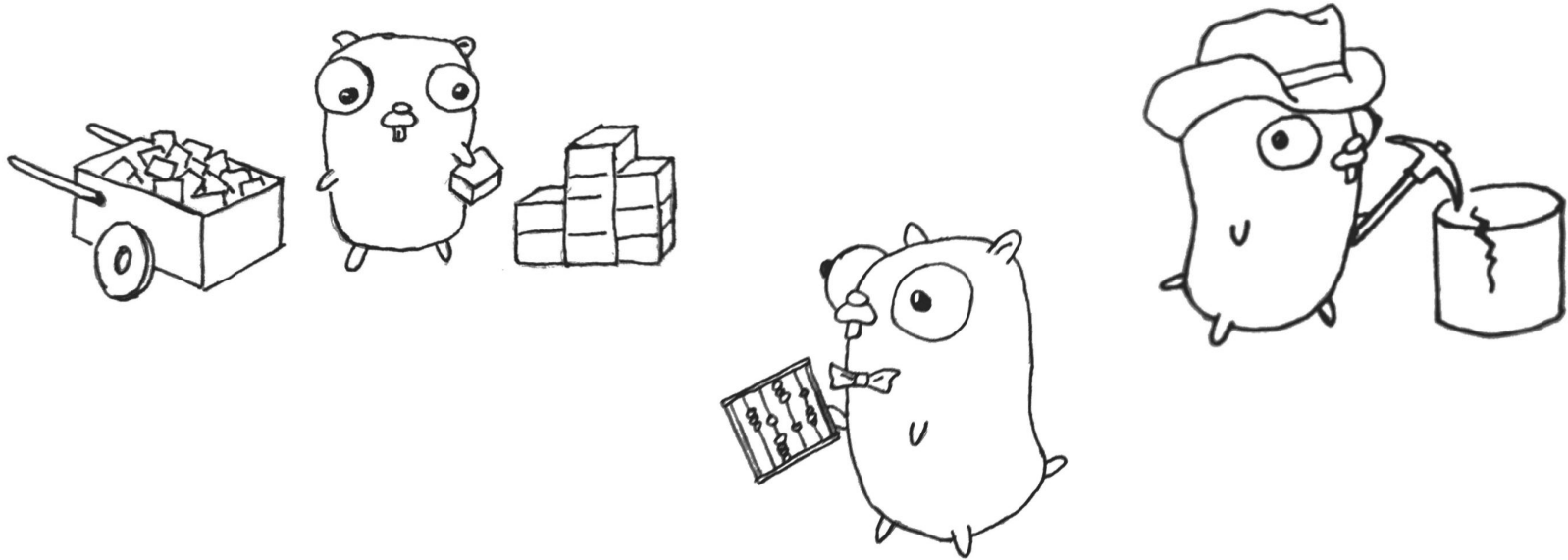
```
FROM scratch  
ADD myservice /myservice  
CMD ["/myservice"]
```



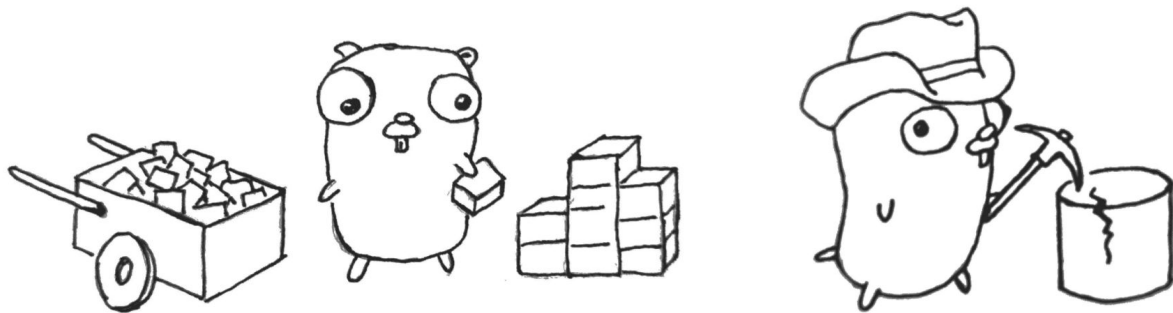


# Data science with Go

Let's put these guys to work on an example project.



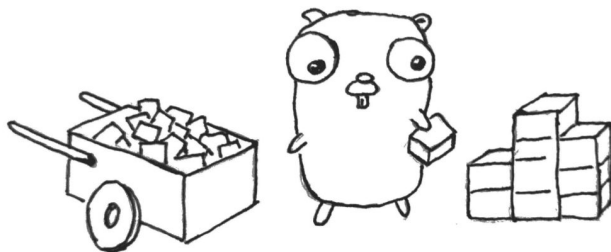
# Data gathering, organizing, parsing



- Scrape data about Go projects off of Github.
- Example [here](#).

# Data gathering, organizing, parsing

But can't we do better  
than outputting to a  
local CSV?



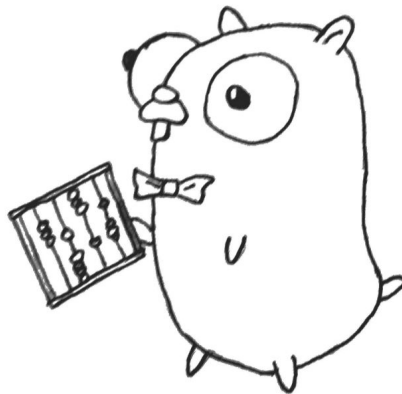
# Data gathering, organizing, parsing



Yes we can! Use [pachyderm](#), example [here](#).

# Arithmetic, Visualization

Need some help?

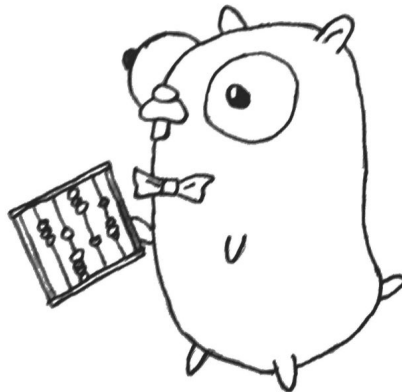


# Arithmetic, Visualization

Need some help?



Nay, I got this!





# Arithmetic, Visualization



[github.com/gonum](https://github.com/gonum)

# Arithmetic, Visualization

[github.com/gonum/plot](https://github.com/gonum/plot)

[github.com/gonum/graph](https://github.com/gonum/graph)

[github.com/gonum/stat](https://github.com/gonum/stat)

[github.com/gonum/integrate](https://github.com/gonum/integrate)

[github.com/gonum/lapack](https://github.com/gonum/lapack)

[github.com/gonum/unit](https://github.com/gonum/unit)

[github.com/gonum/matrix](https://github.com/gonum/matrix)

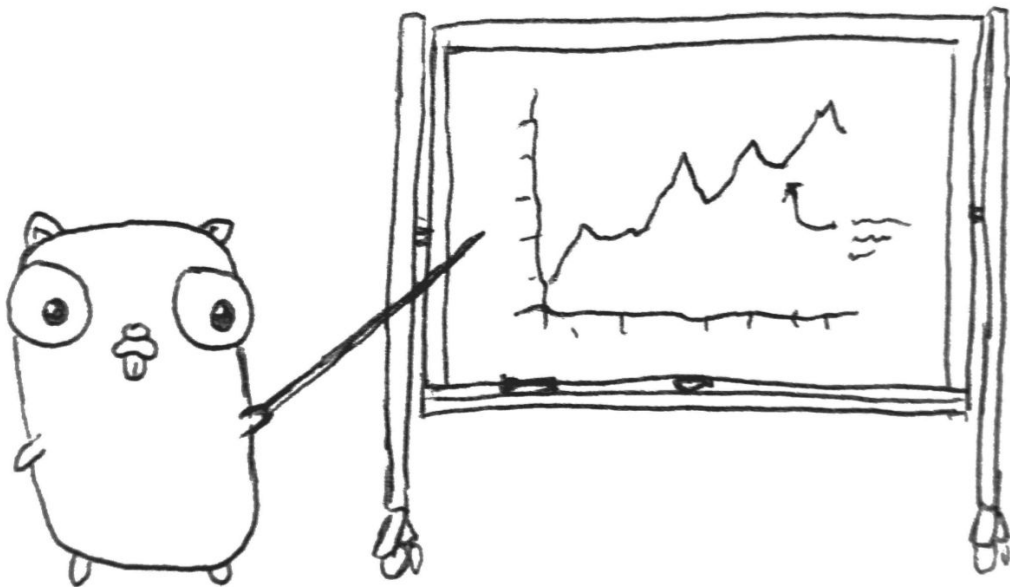
[github.com/gonum/mathext](https://github.com/gonum/mathext)

[github.com/gonum/floats](https://github.com/gonum/floats)

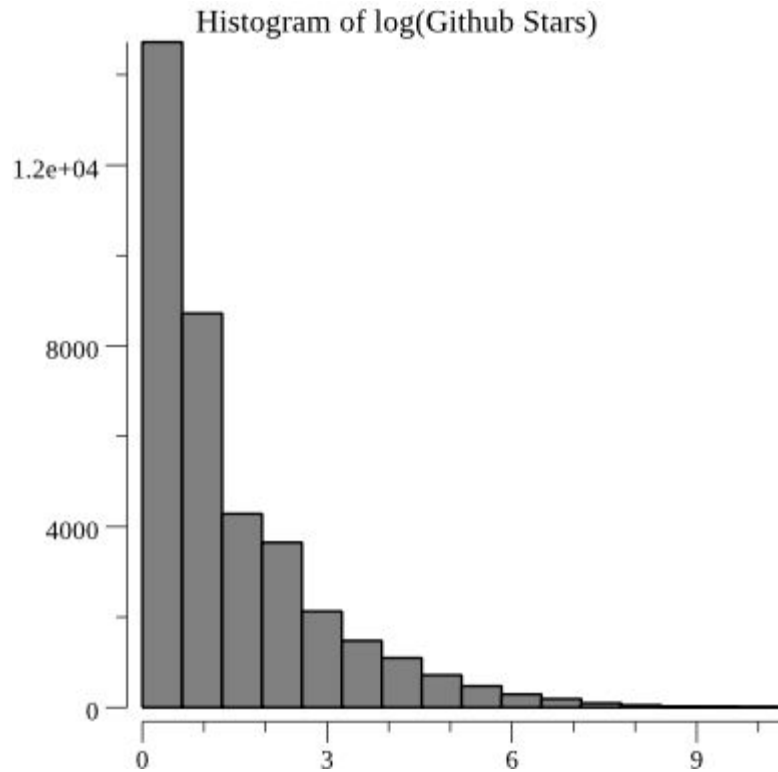
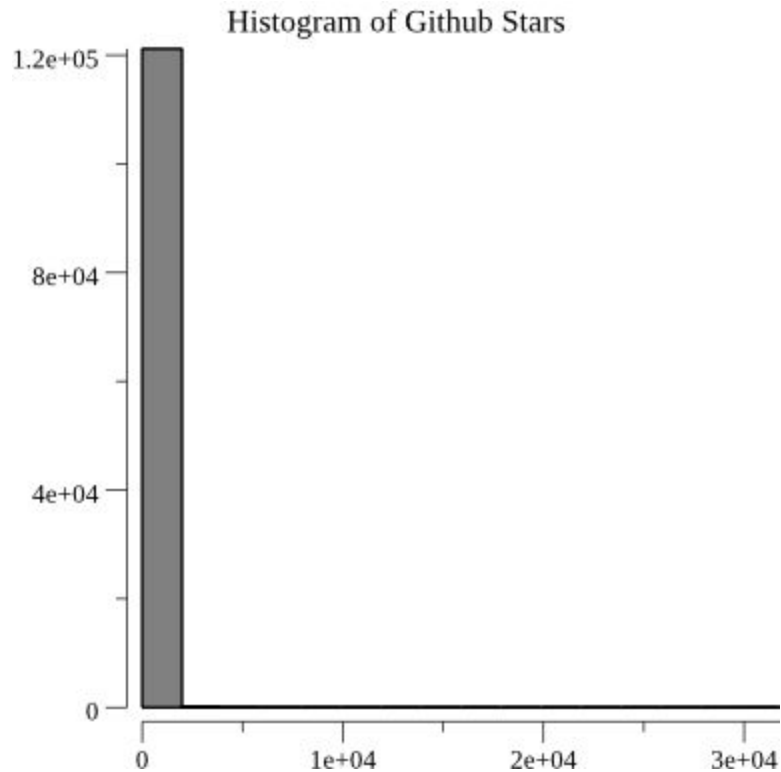
[github.com/gonum/blas](https://github.com/gonum/blas)

[github.com/gonum/optimize](https://github.com/gonum/optimize)

# Arithmetic, Visualization



[github.com/gonum/plot](https://github.com/gonum/plot)



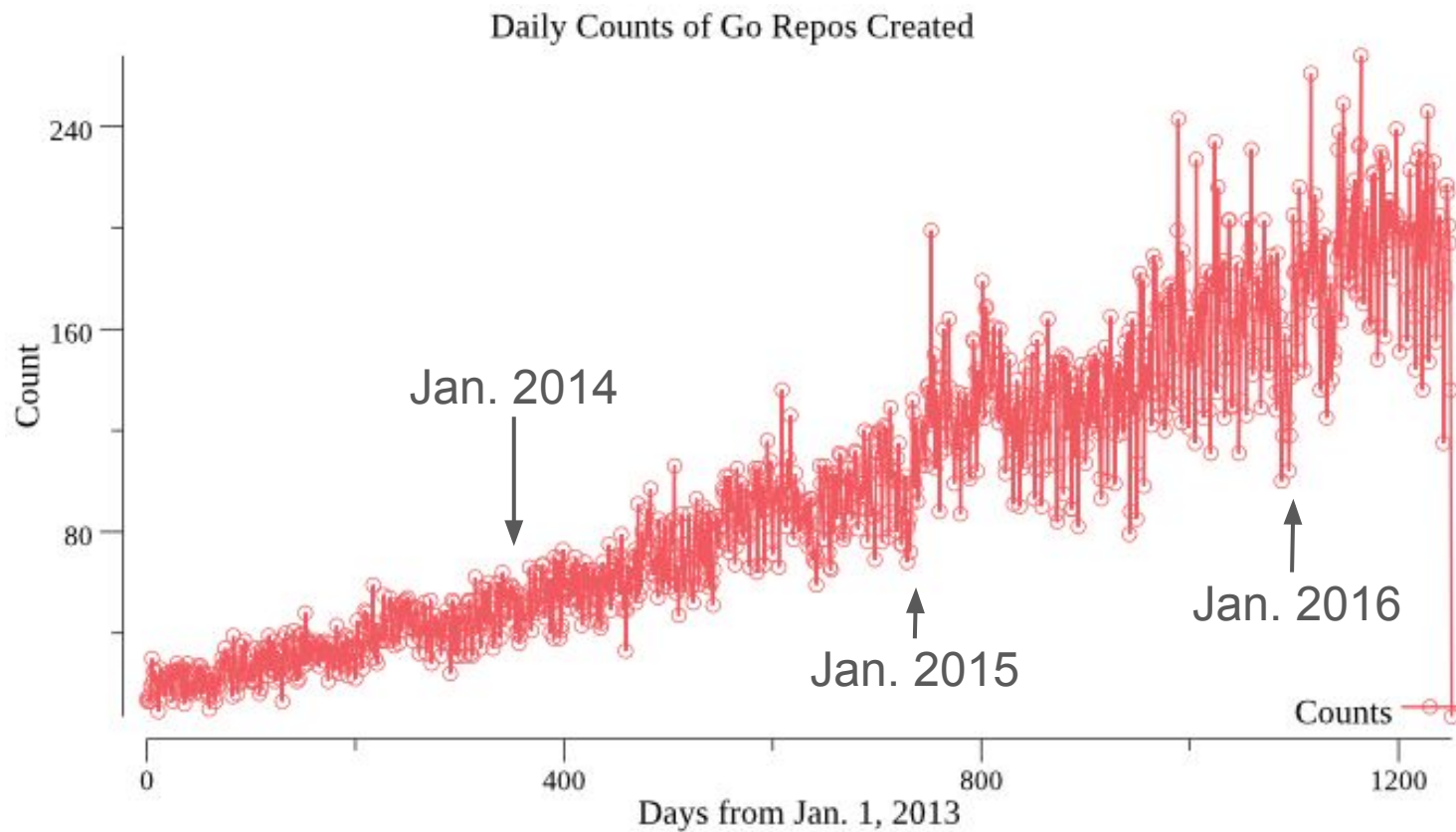
Example [here](#).

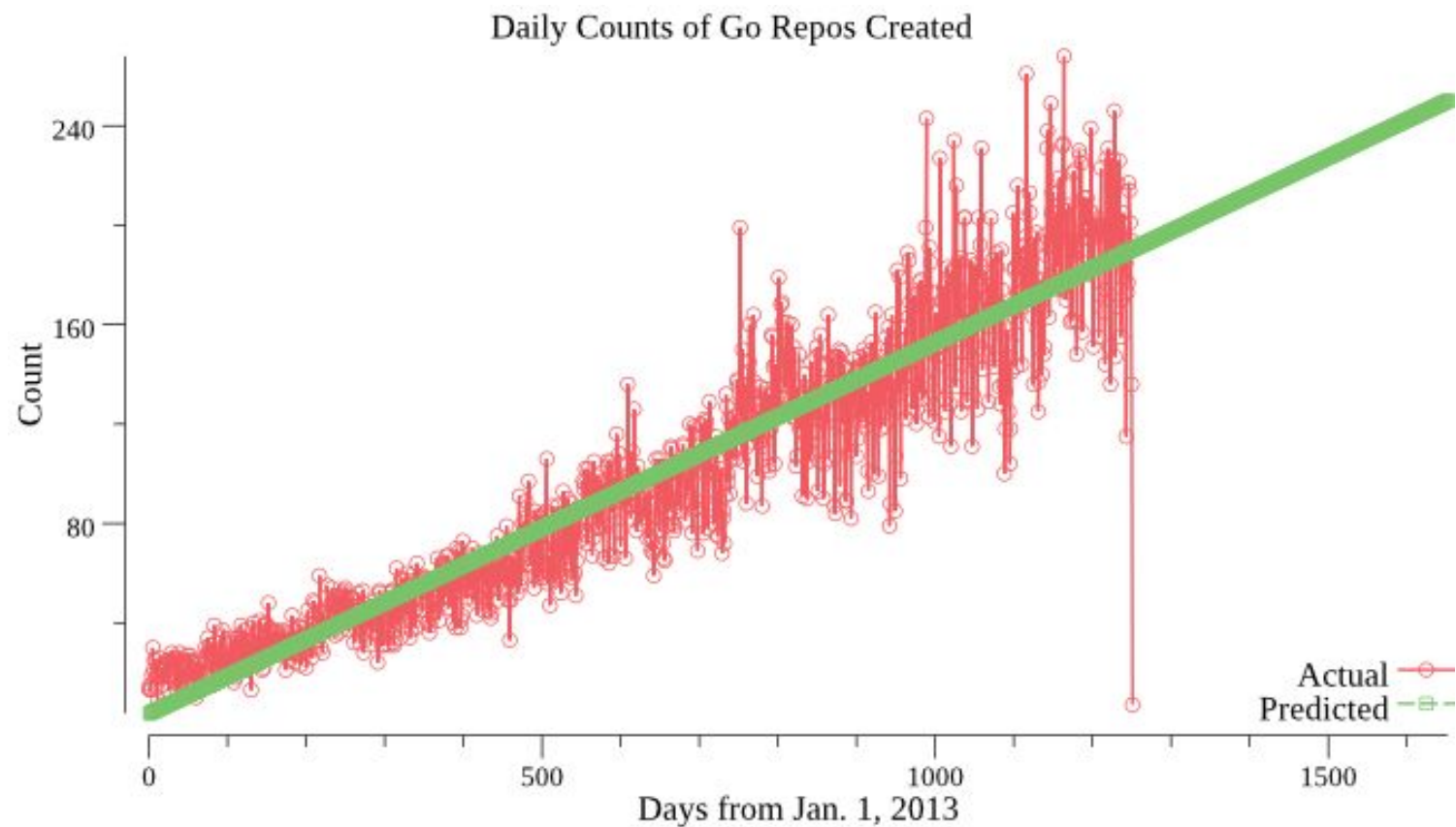
# Prediction

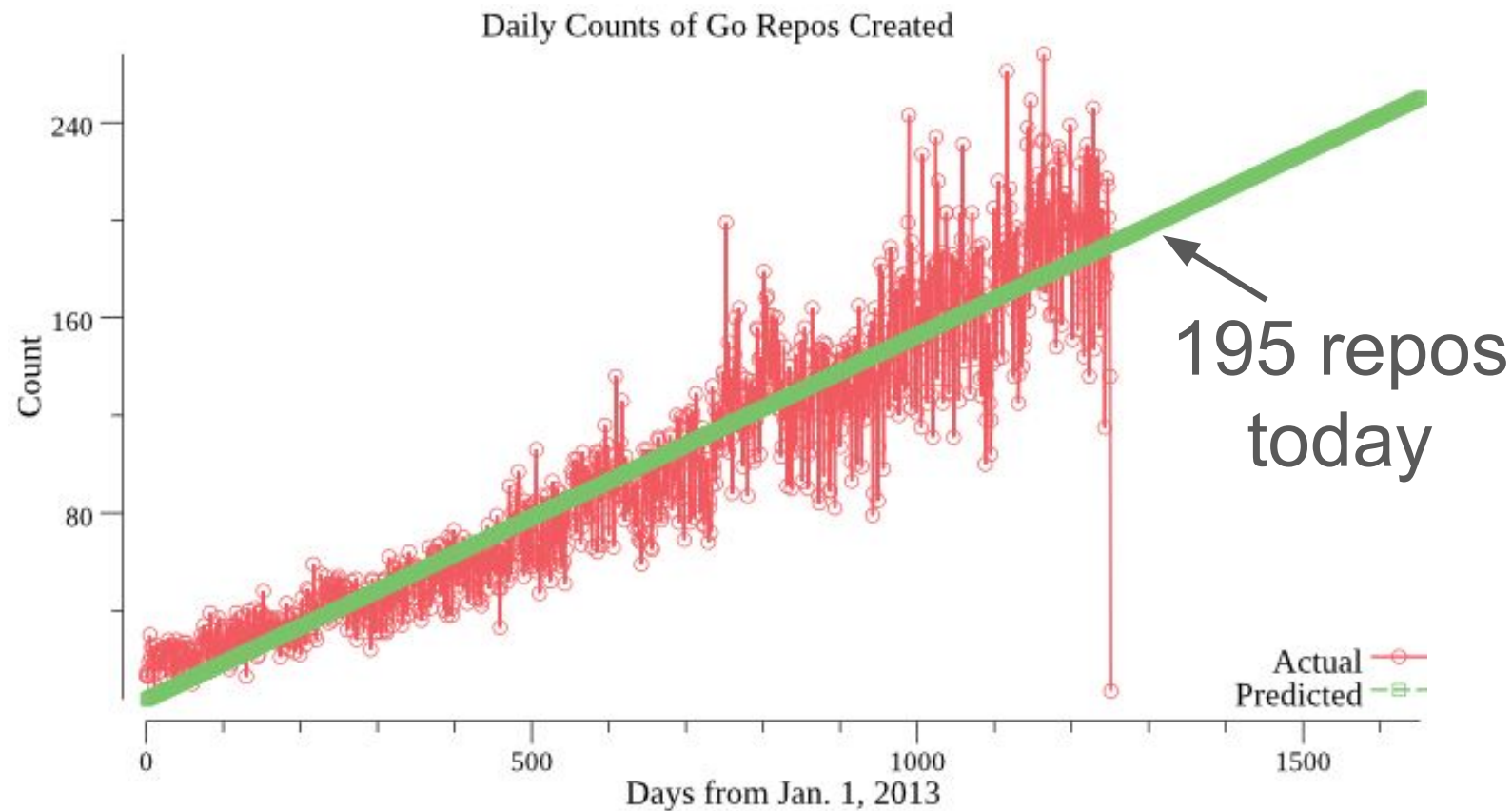


Example [here](#), using

`github.com/sajari/regression`

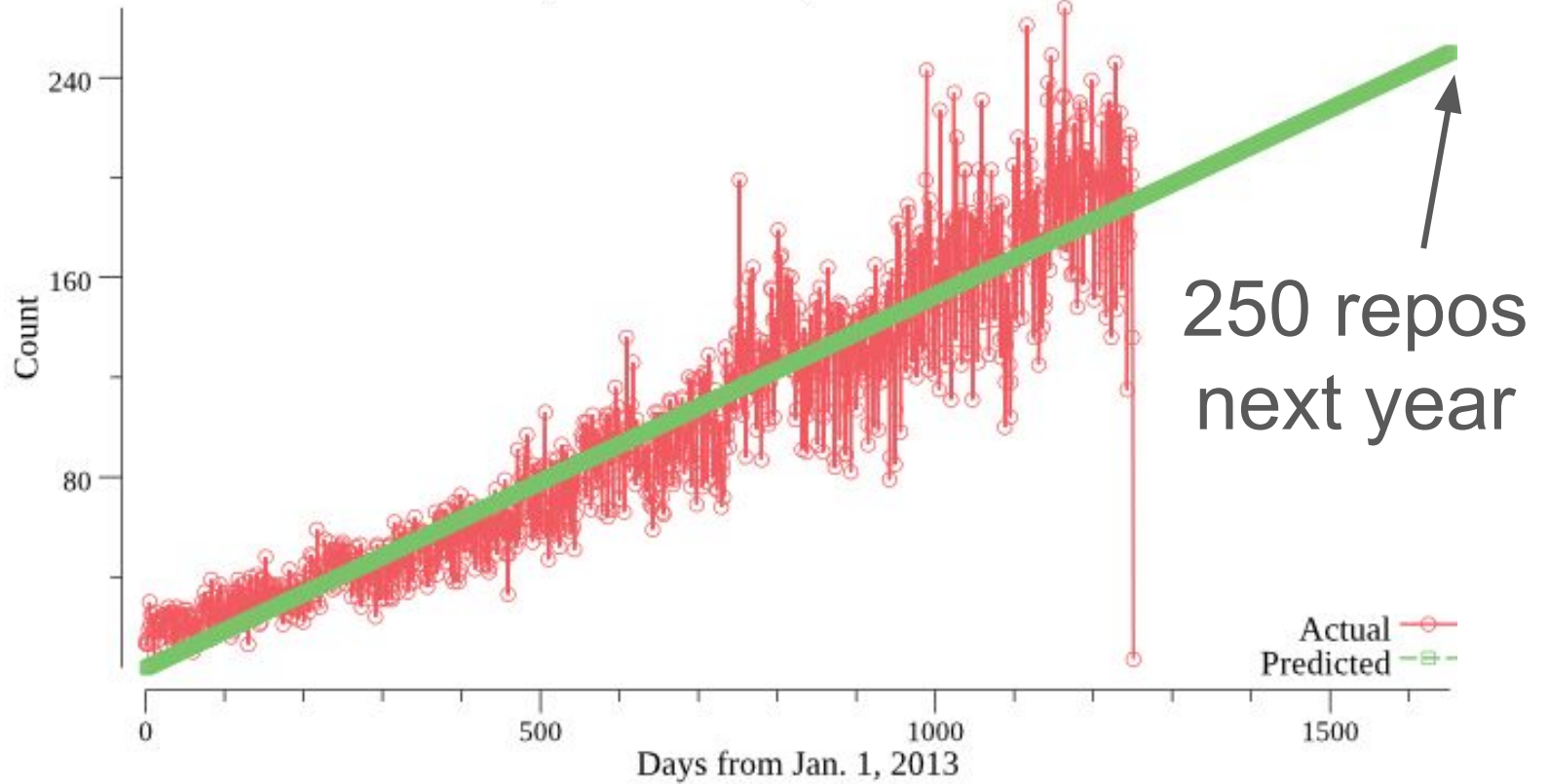








Daily Counts of Go Repos Created



# Get started, Contribute

# Data-Related Go Projects Used Here

- [github.com/gonum](https://github.com/gonum) - numerical computing, statistics, visualization
- [github.com/pachyderm/pachyderm](https://github.com/pachyderm/pachyderm) - containerized data pipelines and data versioning
- [github.com/sajari/regression](https://github.com/sajari/regression) - multivariate regression
- [github.com/pkg/errors](https://github.com/pkg/errors) - error handling

# Other Data-Related Go Projects

- [gophernotes](#) - interactive Go in notebooks
- [GoLearn](#) - general purpose ML
- [glow](#) - distributed computation system (map reduce)
- [GoBot](#) - Go for sensors, IoT
- [gota](#) - DataFrames and data wrangling
- Many, many more listed [here](#)

# Already on the Roadmap

- golang/go proposal for multidimensional slices/tables (see [here](#)).
- Go API for TensorFlow.



# Thank you!

Daniel Whitenack, [Ardan Labs](#)  
[datadan.io](#), [@dwhitena](#), dwhitena on gophers slack

The Go Gopher, the basis of the drawings herein, was created by [Renée French](#).