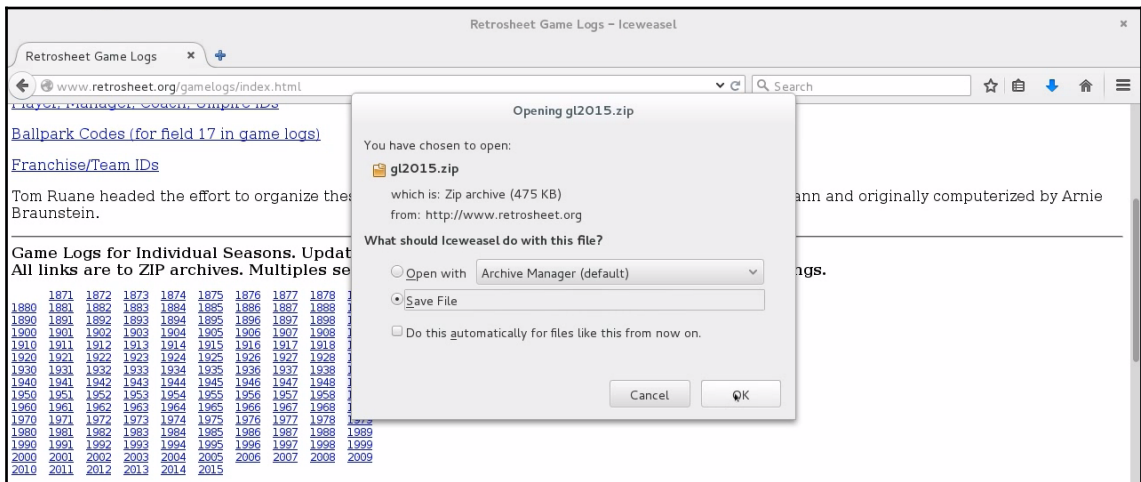
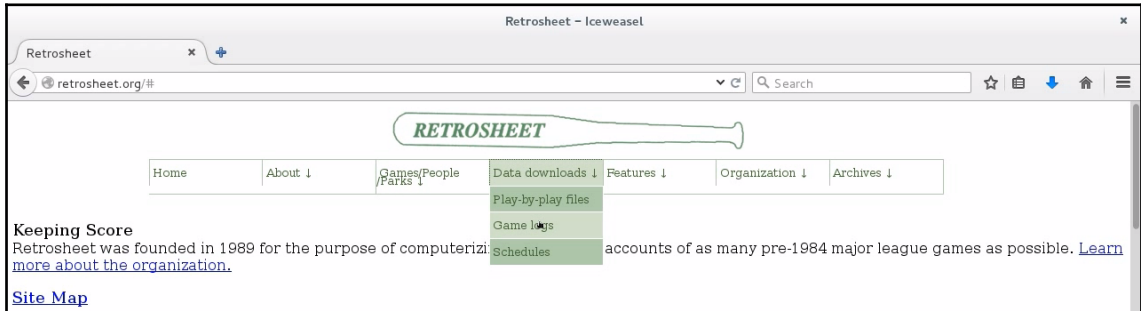


Chapter 1: Descriptive Statistics



```
jcchurch@dataanalysis: ~/Downloads
File Edit View Search Terminal Help
jcchurch@dataanalysis:~$ cd Downloads/
jcchurch@dataanalysis:~/Downloads$ ls
gl2015.zip
jcchurch@dataanalysis:~/Downloads$ unzip gl2015.zip
Archive: gl2015.zip
  inflating: GL2015.TXT
jcchurch@dataanalysis:~/Downloads$ vi GL2015.TXT
```

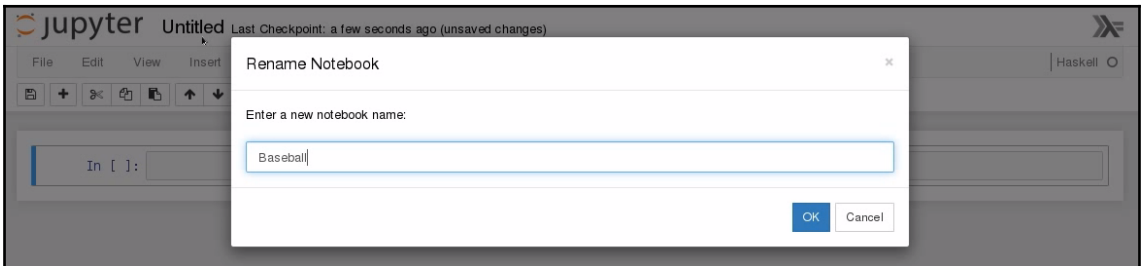
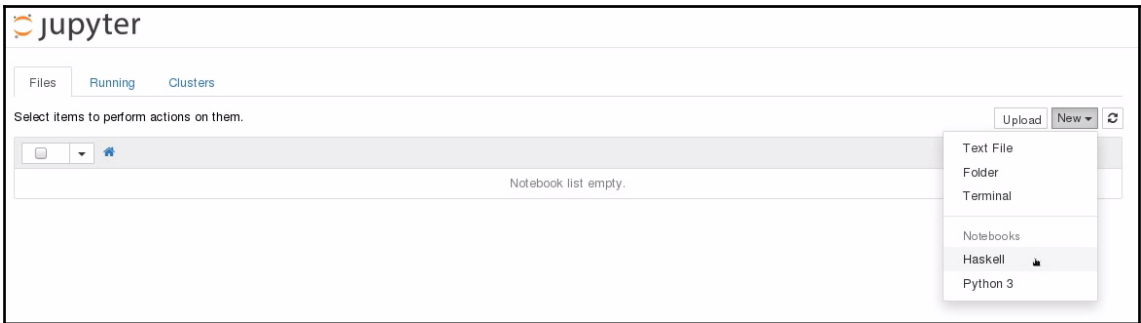
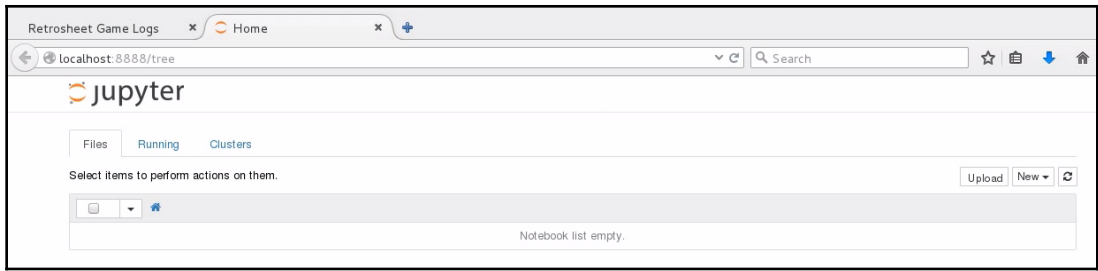
```
jchurch@dataanalysis: ~/Downloads
File Edit View Search Terminal Help
"20150405", "0", "Sun", "SLN", "NL", 1, "CHN", "NL", 1, 3, 0, 54, "N", "", "", "", "CHI11", 35055
, 184, "110010000", "000000000", 36, 10, 3, 0, 0, 3, 0, 0, 0, 4, 1, 11, 4, 1, 0, 0, 10, 4, 0, 0, 0, 0, 27,
8, 0, 1, 0, 0, 32, 5, 3, 0, 0, 0, 0, 0, 2, 0, 12, 1, 0, 0, 0, 7, 6, 3, 3, 0, 0, 27, 8, 2, 0, 0, 0, "wintm901",
"Mike Winters", "wegnm901", "Mark Wegner", "fostm901", "Marty Foster", "muchm901", "Mi
ke Muchlinski", "", "(none)", "", "(none)", "mathm001", "Mike Matheny", "maddj801", "Joe
Maddon", "waina001", "Adam Wainwright", "lestj001", "Jon Lester", "roset001", "Trevor
Rosenthal", "hollm001", "Matt Holliday", "waina001", "Adam Wainwright", "lestj001", "
Jon Lester", "carpm002", "Matt Carpenter", 5, "heywj001", "Jason Heyward", 9, "hollm001
", "Matt Holliday", 7, "peraj001", "Jhonny Peralta", 6, "adamm002", "Matt Adams", 3, "mol
iy001", "Yadier Molina", 2, "wongk001", "Kolten Wong", 4, "jay-j001", "Jon Jay", 8, "wain
a001", "Adam Wainwright", 1, "fowld001", "Dexter Fowler", 8, "solej001", "Jorge Soler",
9, "rizza001", "Anthony Rizzo", 3, "casts001", "Starlin Castro", 6, "coghc001", "Chris C
oghlan", 7, "olt-m001", "Mike Olt", 5, "rossd001", "David Ross", 2, "lestj001", "Jon Lest
er", 1, "lastt001", "Tommy La Stella", 4, "", "Y"^\M
@
@
@
@
@
@
@
@
"GL2015.TXT" 2429 lines, 2648768 characters
```

```
"20150405", "0", "Sun", "SLN", "NL", 1, "CHN", "NL", 1, 3, 0, 54, "N", "", "", "", "CHI11", 35055
, 184, "110010000", "000000000", 36, 10, 3, 0, 0, 3, 0, 0, 0, 4, 1, 11, 4, 1, 0, 0, 10, 4, 0, 0, 0, 0, 27,
8, 0, 1, 0, 0, 32, 5, 3, 0, 0, 0, 0, 0, 2, 0, 12, 1, 0, 0, 0, 7, 6, 3, 3, 0, 0, 27, 8, 2, 0, 0, 0, "wintm901",
"Mike Winters", "wegnm901", "Mark Wegner", "fostm901", "Marty Foster", "muchm901", "Mi
ke Muchlinski", "", "(none)", "", "(none)", "mathm001", "Mike Matheny", "maddj801", "Joe
Maddon", "waina001", "Adam Wainwright", "lestj001", "Jon Lester", "roset001", "Trevor
Rosenthal", "hollm001", "Matt Holliday", "waina001", "Adam Wainwright", "lestj001", "
Jon Lester", "carpm002", "Matt Carpenter", 5, "heywj001", "Jason Heyward", 9, "hollm001
", "Matt Holliday", 7, "peraj001", "Jhonny Peralta", 6, "adamm002", "Matt Adams", 3, "mol
iy001", "Yadier Molina", 2, "wongk001", "Kolten Wong", 4, "jay-j001", "Jon Jay", 8, "wain
a001", "Adam Wainwright", 1, "fowld001", "Dexter Fowler", 8, "solej001", "Jorge Soler",
9, "rizza001", "Anthony Rizzo", 3, "casts001", "Starlin Castro", 6, "coghc001", "Chris C
oghlan", 7, "olt-m001", "Mike Olt", 5, "rossd001", "David Ross", 2, "lestj001", "Jon Lest
er", 1, "lastt001", "Tommy La Stella", 4, "", "Y"^\M
```

```
jcchurch@dataanalysis:~/Downloads$ cabal install csv
Resolving dependencies...
cabal: Entering directory '/tmp/cabal-tmp-8375/csv-0.1.2'
Configuring csv-0.1.2...
Building csv-0.1.2...
Preprocessing library csv-0.1.2...
[1 of 1] Compiling Text.CSV          ( Text/CSV.hs, dist/build/Text/CSV.o )
Creating package registration file: /tmp/pkgConf-csv-0.18375.2
Installing library in
/home/jcchurch/.cabal/lib/x86_64-linux-ghc-7.6.3/csv-0.1.2-HemIOijiMwyGSAMWRD0xZ
A
Registering csv-0.1.2...
cabal: Leaving directory '/tmp/cabal-tmp-8375/csv-0.1.2'
Installed csv-0.1.2
jcchurch@dataanalysis:~/Downloads$ █
```

```
jcchurch@dataanalysis:~/Downloads$ cd ..
jcchurch@dataanalysis:~$ ls
Code      Documents  Dropbox   Pictures  Templates
Desktop  Downloads  Music     Public    Videos
jcchurch@dataanalysis:~$ cd Code/
jcchurch@dataanalysis:~/Code$ ls
HaskellDataAnalysis  PreliminaryWork
jcchurch@dataanalysis:~/Code$ cd HaskellDataAnalysis/
jcchurch@dataanalysis:~/Code/HaskellDataAnalysis$ ls
analysis  data
jcchurch@dataanalysis:~/Code/HaskellDataAnalysis$ cp ~/Downloads/GL2015.TXT data/
jcchurch@dataanalysis:~/Code/HaskellDataAnalysis$ ls data/
GL2015.TXT
```

```
jcchurch@dataanalysis:~/Code/HaskellDataAnalysis$ cd analysis/
jcchurch@dataanalysis:~/Code/HaskellDataAnalysis/analysis$ ls
jcchurch@dataanalysis:~/Code/HaskellDataAnalysis/analysis$ jupyter notebook
/usr/local/lib/python3.4/dist-packages/widgetsnbextension/__init__.py:30: UserWarning: To use the jupyter-js-widgets nbextension, you'll need to update
the Jupyter notebook to version 4.2 or later.
the Jupyter notebook to version 4.2 or later."")
[I 15:51:06.048 NotebookApp] Serving notebooks from local directory: /home/jcchurch/Code/HaskellDataAnalysis/analysis
[I 15:51:06.052 NotebookApp] 0 active kernels
[I 15:51:06.053 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/
[I 15:51:06.053 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```



```
In [2]: baseball <- parseCSVFromFile "../data/GL2015.TXT"
```

```
In [ ]:
```

```
In [3]: :type baseball
```

```
baseball :: Either ParseError CSV
```

```
In [4]: noEmptyRows = either (const []) (filter (\row -> 2 <= length row))
```

```
In [5]: baseballList = noEmptyRows baseball
```

```
In [6]: length baseballList
```

```
2429
```

```
In [7]: :type baseballList
```

```
baseballList :: [[Field]]
```

```
In [8]: :info Field
```

```
In [ ]: |
```

```
type Field = String -- Defined in `Text.CSV`
```

```
In [9]: "1" + 1
```

```
No instance for (Num String) arising from a use of `+'  
Possible fix: add an instance declaration for (Num String)  
In the expression: "1" + 1  
In an equation for `it': it = "1" + 1
```

```
In [10]: read "1" :: Integer
```

```
1
```

```
In [11]: read "1.5" :: Double
```

```
1.5
```

```
In [12]: readIndex :: Read cell => Either a CSV -> Int -> [cell]
         readIndex csv index = map (read . (!! index)) (noEmptyRows csv)
```

```
In [13]: readIndex baseball 9 :: [Integer]
```

```
[3,0,0,1,6,0,1,6,5,2,3,2,10,8,3,3,2,6,6,7,12,5,0,2,5,3,0,5,0,5,0,4,4,2,5,2,1,1,5,1,6,10,2,6,1,6,4,12,6,8,6,0,5,3,3,4,1,9,
6,1,0,6,1,4,9,8,5,2,0,3,4,9,2,0,2,2,9,10,2,8,4,8,6,7,4,7,6,8,10,4,4,6,4,8,12,6,2,2,6,5,0,4,8,2,5,3,7,4,4,2,3,8,3,5,5,2,1,
4,5,10,2,1,1,10,7,6,0,2,1,1,2,4,2,5,4,5,7,0,2,2,1,6,4,2,3,5,8,5,3,1,3,9,1,2,4,12,0,5,4,1,9,5,6,3,4,2,1,2,5,8,1,3,2,2,10,5
,5,5,0,6,2,5,1,1,6,1,3,1,1,7,14,6,5,1,6,5,5,6,1,6,7,7,16,1,3,9,2,1,9,0,13,3,2,5,2,5,4,2,2,6,3,2,7,0,3,2,1,6,1,2,3,9,4,2,4
,2,7,2,13,1,5,0,3,4,7,4,2,3,0,3,1,4,1,8,9,8,2,2,5,0,5,5,11,5,7,3,6,4,7,4,1,8,5,2,3,4,1,5,6,5,1,3,5,4,0,6,3,3,9,4,11,11,2
,2,2,5,13,2,2,2,3,14,5,2,1,5,10,3,6,5,1,13,8,8,3,3,7,2,5,2,1,2,6,5,8,3,2,3,4,3,1,0,7,3,0,0,3,0,3,2,1,0,4,11,4,2,3,7,8,6
,4,0,1,2,4,1,2,8,7,6,6,3,7,0,5,0,6,1,6,0,2,3,5,2,7,1,5,3,0,9,4,4,0,2,7,3,2,6,0,8,2,7,0,4,2,3,5,6,11,10,0,1,5,13,5,3,1,3,9
,6,5,3,4,4,5,3,5,0,14,2,7,1,0,9,5,4,3,2,0,6,2,7,1,8,0,2,6,10,2,7,6,6,2,2,1,6,4,3,5,6,6,1,3,2,2,2,3,2,6,1,9,2,7,3,2,4,2,5,1
1,2,11,3,2,3,7,4,2,10,8,1,8,2,4,2,7,6,1,3,1,4,7,1,1,0,6,3,2,4,2,2,9,1,1,5,4,2,2,1,4,2,1,6,5,3,5,2,3,3,4,1,2,7,1,8,10,10,4
,5,7,3,10,10,6,5,5,4,4,4,10,1,11,7,5,1,5,4,4,0,2,0,11,7,0,1,3,9,0,6,1,0,10,1,1,3,2,6,4,3,1,4,3,3,8,4,0,3,5,5,4,10,8,3,0,6
,4,2,4,2,1,1,4,1,4,6,9,4,3,0,2,4,3,5,5,0,4,1,3,7,0,3,0,12,2,3,2,0,10,2,4,4,11,11,1,8,1,1,3,4,1,3,2,15,5,3,9,2,10,3,0,0,2
,8,1,8,2,10,6,5,7,2,3,1,2,11,2,1,1,3,3,10,2,1,0,4,0,2,5,3,8,3,2,2,4,4,4,1,1,1,7,9,2,1,0,6,4,1,4,5,4,3,4,2,3,3,5,3,6,3,3,0
,2,3,2,3,3,4,5,5,11,0,0,1,6,6,2,1,4,8,2,7,4,4,2,2,0,6,3,0,2,5,4,0,5,7,9,5,5,8,5,2,9,6,5,0,6,3,1,2,6,3,4,1,7,1,3,2,7,8,11,5
,7,4,1,6,0,5,4,2,5,2,6,3,9,2,4,2,7,0,0,7,6,3,2,6,1,2,3,9,8,6,3,4,3,5,4,8,8,7,3,6,2,1,6,7,6,2,2,3,5,4,10,7,1,2,2,10,6,6,2
,5,5,2,7,1,4,4,2,1,2,1,4,9,5,0,7,4,4,6,7,3,0,2,3,6,6,3,0,3,4,4,6,1,3,3,5,4,3,3,2,0,2,3,0,2,1,2,8,3,5,2,3,1,5,4,2,1,9,12,7
,5,4,2,2,1,2,4,6,8,0,5,0,0,6,6,3,0,5,4,4,3,13,0,0,5,2,5,1,4,3,0,4,1,0,0,4,5,5,8,4,7,3,1,7,5,3,1,4,2,8,5,13,1,0,1,4,1,4,4,8
,0,4,4,7,0,4,0,3,1,1,1,8,3,0,9,5,2,1,3,4,5,5,16,2,6,2,7,2,0,6,2,2,3,17,2,1,1,2,0,0,2,2,4,8,5,6,5,3,3,2,1,4,3,7,1,7,5,8,0
,1,5,2,4,7,2,2,12,2,4,2,1,0,9,9,12,1,2,4,4,4,3,1,3,5,8,4,5,1,6,10,0,2,0,13,8,12,2,6,13,2,0,2,4,2,2,2,3,8,2,11,4,8,2,13,6,7
,6,11,0,3,8,0,5,4,2,6,3,1,1,1,2,1,2,8,1,8,5,8,6,1,5,0,1,8,8,0,6,4,4,5,2,5,8,0,3,3,4,3,5,4,2,7,4,1,5,2,2,8,2,2,9,3,1,4,2,5
```



```
In [14]: awayRuns = readIndex baseball 9 :: [Integer]
```

```
In [15]: maximum awayRuns
         21

In [16]: minimum awayRuns
         0
```

```
In [17]: :type (minimum, maximum)
         (minimum, maximum) :: forall a1. (Ord a, Ord a1) => ([a] -> a, [a1] -> a1)
```

```
In [18]: range :: Ord a => [a] -> (a, a)
         range xs = (minimum xs, maximum xs)
```

```
In [19]: range awayRuns
         (0,21)
```

```
In [20]: range []
         Prelude.minimum: empty list
```

```
In [21]: range :: Ord a => [a] -> Maybe (a, a)
         range [] = Nothing
         range [x] = Just (x, x)
         range xs = Just (minimum xs, maximum xs)
```

```
In [22]: range []
         Nothing
```

```
In [23]: range [1]
         Just (1,1)
```

```
In [24]: range awayRuns
         Just (0,21)
```

$$\text{mean}(X) = \bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2}$$

```
In [25]: import Text.CSV
         import Data.Maybe
```

```
In [25]: sum awayRuns
```

```
10091
```

```
In [26]: length awayRuns
```

```
2429
```

```
In [27]: :type sum
```

```
sum :: forall a. Num a => [a] -> a
```

```
In [28]: :type length
```

```
length :: forall a. [a] -> Int
```

```
In [29]: realToFrac (sum awayRuns) / fromIntegral (length awayRuns)
```

```
4.154384520378756
```

```
In [30]: mean :: Real a => [a] -> Maybe Double
```

```
mean [] = Nothing
```

```
mean [x] = Just $ realToFrac x
```

```
mean xs = Just $ realToFrac (sum xs) / fromIntegral (length xs)
```

```
In [31]: mean []
```

```
Nothing
```

```
In [32]: mean [1]
```

```
Just 1.0
```

```
In [33]: mean awayRuns
```

```
Just 4.154384520378756
```

```
In [34]: stdev :: Real a => [a] -> Maybe Double
```

```
stdev [] = Nothing
```

```
stdev [_] = Nothing
```

```
stdev xs = Just $ sqrt (sumsquares / n_m1)
```

```
where
```

```
  n_m1 = fromIntegral (length xs - 1)
```

```
  meanxs = fromJust (mean xs)
```

```
  sumsquares = sum $ map (diffsquare . realToFrac) xs
```

```
  diffsquare x = (x - meanxs) * (x - meanxs)
```



```
In [35]: stdev []  
Nothing  
  
In [36]: stdev [1]  
Nothing  
  
In [37]: stdev awayRuns  
Just 3.1155073817635124
```

```
In [38]: (4.15 - 3.12, 4.15 + 3.12)  
(1.0300000000000002, 7.2700000000000005)
```

```
In [*]: import Text.CSV  
import Data.Maybe  
import Data.List
```

```
In [39]: oddList = [3,4,1,2,5]  
  
In [40]: evenList = [6,5,4,3,2,1]
```

```
In [41]: sort oddList  
[1,2,3,4,5]  
  
In [42]: sort evenList  
[1,2,3,4,5,6]
```

```
In [43]: length oddList `div` 2  
2
```

```
In [44]: sort oddList !! 2  
3
```

```
In [45]: length evenList `div` 2  
3
```

```
In [46]: ((sort evenList !! 3) + (sort evenList !! 2)) / 2
3.5
```

```
In [47]: median :: Real a => [a] -> Maybe Double
median [] = Nothing
median list
  | odd (length list) = Just middleValue
  | otherwise = Just middleEven
  where
    sorted = sort list
    middleIndex = length list `div` 2
    middleValue = realToFrac $ sorted !! middleIndex
    beforeMiddleValue = realToFrac $ sorted !! (middleIndex - 1)
    middleEven = 0.5 * (middleValue + beforeMiddleValue)
```

```
In [48]: median []
Nothing

In [49]: median oddList
Just 3.0

In [50]: median evenList
Just 3.5
```

```
In [51]: median awayRuns
Just 4.0
```

```
In [1]: import Text.CSV
import Data.Maybe
import Data.List
import Data.Ord
```

```
In [52]: myList = [4,4,5,5,4]
```

```
In [53]: group myList
[[4,4],[5,5],[4]]
```

```
In [54]: runLengthEncoding :: Ord a => [a] -> [(a, Integer)]
runLengthEncoding = (map (\xs -> (head xs, genericLength xs)) . group)
```

```
In [55]: runLengthEncoding myList
        [(4,2),(5,2),(4,1)]
```

```
In [56]: runLengthEncoding []
        []
```

```
In [57]: runLengthEncoding (sort myList)
        [(4,3),(5,2)]
```

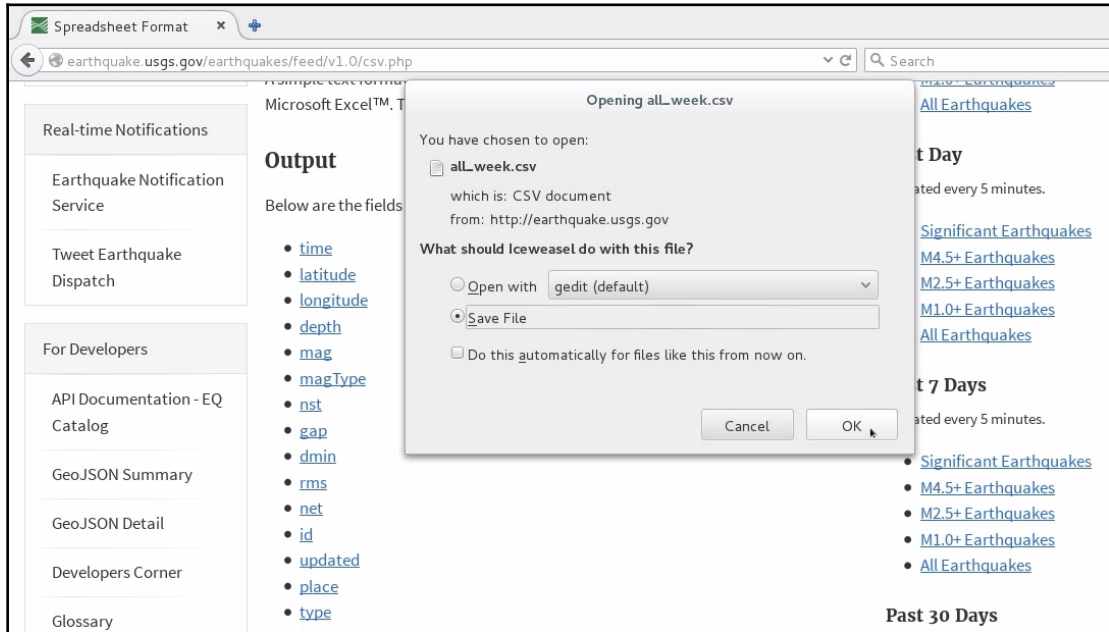
```
In [58]: mode :: Ord a => [a] -> Maybe (a, Integer)
mode [] = Nothing
mode list = Just $ maximumBy (comparing snd) pairs
  where
    sorted = sort list
    pairs = runLengthEncoding sorted
```

```
In [59]: mode []
        Nothing
```

```
In [60]: mode myList
        Just (4,3)
```

```
In [61]: mode awayRuns
        Just (2,379)
```

Chapter 2: SQLite3



```
File Edit View Search Terminal Help
jcchurch@dataanalysis:~/Downloads$ ls
all_week.csv  GL2015.TXT  gl2015.zip
jcchurch@dataanalysis:~/Downloads$
```

```
File Edit View Search Terminal Help
jcchurch@dataanalysis:~/Downloads$ ls
all_week.csv  GL2015.TXT  gl2015.zip
jcchurch@dataanalysis:~/Downloads$ vi all_week.csv
```

```
File Edit View Search Terminal Help
time,latitude,longitude,depth,mag,magType,nst,gap,dmin,rms,net,id,updated,place,
type,horizontalError,depthError,magError,magNst,status,locationSource,magSource
2016-07-11T19:44:16.470Z,38.2513,-115.9473,20.2,1.7,m1,11,176.49,1.034,,nn,nn005
51404,2016-07-11T19:46:20.977Z,"37km E of Warm Springs, Nevada",earthquake,,,,a
utomatic,nn,nn
2016-07-11T19:38:52.140Z,38.2467,-118.6373,7,0.8,m1,6,200.49,0.095,,nn,nn0055140
3,2016-07-11T19:41:07.909Z,"30km S of Hawthorne, Nevada",earthquake,,,,automati
c,nn,nn
2016-07-11T19:27:19.750Z,38.6712,-117.0597,0.3,1.7,m1,9,159.2,0.888,,nn,nn005514
02,2016-07-11T19:29:35.169Z,"68km NNE of Tonopah, Nevada",earthquake,,,,automat
ic,nn,nn
2016-07-11T19:01:47.820Z,48.0008333,-121.9131667,11.92,1.8,m1,12,93,0.2198,0.18,
uw,uw61179846,2016-07-11T19:15:53.980Z,"9km NE of Three Lakes, Washington",earth
quake,0.44,2.69,0.107,20,reviewed,uw,uw
2016-07-11T18:48:12.000Z,60.2407,-141.6943,13.7,1.3,m1,,,0.49,ak,ak13751113,201
6-07-11T19:07:17.191Z,"45km ENE of Cape Yakataga, Alaska",earthquake,0.2,0.1,,,a
utomatic,ak,ak
2016-07-11T18:46:07.360Z,33.695,-117.4286667,5.76,1.14,m1,29,86,0.04586,0.2,ci,c
i37412463,2016-07-11T18:49:57.670Z,"10km WNW of Lake Elsinore, CA",earthquake,0.
33,0.79,0.165,29,automatic,ci,ci
2016-07-11T18:44:07.370Z,47.166,-121.9456667,-1.06,0.94,m1,8,121,0.03681,0.18,uw
,uw61179836,2016-07-11T19:11:40.350Z,"5km SE of Enumclaw, Washington",explosion,
0.68,31.61,0.049,5,reviewed,uw,uw
"all_week.csv" 1586 lines, 293297 characters
```

```
jcchurch@dataanalysis:~/Downloads$ cut -d, -f 1,2,3,4,5 all_week.csv > earthquak
es.csv
jcchurch@dataanalysis:~/Downloads$ █
```

```
jcchurch@dataanalysis:~/Downloads$ tail -n +2 earthquakes.csv > e.tmp
jcchurch@dataanalysis:~/Downloads$ █
```

```
jcchurch@dataanalysis:~/Downloads$ mv e.tmp earthquakes.csv
jcchurch@dataanalysis:~/Downloads$ █
```

```
jcchurch@dataanalysis:~/Downloads$ vi earthquakes.csv █
```

```
File Edit View Search Terminal Help
2016-07-11T19:44:16.470Z,38.2513,-115.9473,20.2,1.7
2016-07-11T19:38:52.140Z,38.2467,-118.6373,7,0.8
2016-07-11T19:27:19.750Z,38.6712,-117.0597,0.3,1.7
2016-07-11T19:01:47.820Z,48.0008333,-121.9131667,11.92,1.8
2016-07-11T18:48:12.000Z,60.2407,-141.6943,13.7,1.3
2016-07-11T18:46:07.360Z,33.695,-117.4286667,5.76,1.14
2016-07-11T18:44:07.370Z,47.166,-121.9456667,-1.06,0.94
2016-07-11T18:41:17.000Z,60.6685,-152.2512,95.3,1.2
2016-07-11T18:29:45.000Z,61.2845,-152.5106,0,2.5
2016-07-11T18:02:27.000Z,61.4573,-149.846,52.5,1.3
2016-07-11T17:55:48.650Z,40.4318333,-121.4878333,3.94,1.45
2016-07-11T17:49:03.850Z,40.433,-121.4865,4.13,0.41
2016-07-11T17:42:43.550Z,47.2966667,-122.3013333,16.36,1.67
2016-07-11T17:23:52.610Z,40.4321667,-121.4848333,3.85,0.39
2016-07-11T17:22:30.970Z,49.015,-122.1351667,-0.87,0.79
2016-07-11T17:07:44.040Z,46.5736667,-121.7175,0.25,1.2
2016-07-11T16:59:27.140Z,40.4268333,-121.4921667,2.93,2.07
2016-07-11T16:59:11.110Z,40.4315,-121.488,3.88,0.25
2016-07-11T16:56:00.790Z,33.365,-116.7905,11.82,0.47
2016-07-11T16:54:45.260Z,36.7864,142.1313,29.75,4.7
2016-07-11T16:51:07.420Z,34.0193333,-117.1201667,1.16,1.86
2016-07-11T16:30:08.730Z,45.8876667,-122.439,-0.97,1.23
2016-07-11T16:29:32.450Z,40.432,-121.487,4.13,1.15
"earthquakes.csv" 1585 lines, 84915 characters
```

```
sqlite> CREATE TABLE earthquakes (time TEXT, latitude FLOAT, longitude FLOAT, de  
pth FLOAT, magnitude FLOAT);
```

```
sqlite> .mode csv
sqlite> .import earthquakes.csv earthquakes
```

```
sqlite> SELECT * FROM earthquakes LIMIT 5;
2016-07-11T19:44:16.470Z,38.2513,-115.9473,20.2,1.7
2016-07-11T19:38:52.140Z,38.2467,-118.6373,7,0,0.8
2016-07-11T19:27:19.750Z,38.6712,-117.0597,0.3,1.7
2016-07-11T19:01:47.820Z,48.0008333,-121.9131667,11.92,1.8
2016-07-11T18:48:12.000Z,60.2407,-141.6943,13.7,1.3
sqlite>
```

```
jcchurch@dataanalysis:~/Downloads$ ls -al
total 3552
drwxr-xr-x  2 jcchurch jcchurch   4096 Jul 11 14:58 .
drwxr-xr-x 31 jcchurch jcchurch   4096 Jul 11 14:59 ..
-rw-r--r--  1 jcchurch jcchurch 293297 Jul 11 14:53 all_week.csv
-rw-r--r--  1 jcchurch jcchurch  84915 Jul 11 14:55 earthquakes.csv
-rw-r--r--  1 jcchurch jcchurch 2648768 Dec 30 2015 GL2015.TXT
-rw-r--r--  1 jcchurch jcchurch  486057 Jun 29 15:45 gl2015.zip
-rw-r--r--  1 jcchurch jcchurch 110592 Jul 11 14:58 usgs.sqlite3
jcchurch@dataanalysis:~/Downloads$
```

SQLite3	Haskell
INTEGER	Integer
REAL	Decimal
TEXT	String
BLOB	String

```
jcchurch@dataanalysis:~/Code/HaskellDataAnalysis$ cp ~/Downloads/usgs.sqlite3 data/
```

```

jupyter Earthquakes Last Checkpoint: a few seconds ago (unsaved changes)
File Edit View Insert Cell Kernel Help
[Icons] Code CellToolbar

In [2]: import Data.List
import Database.HDBC.SQLite3
import Database.HDBC

```

```
In [3]: db <- connectSqlite3 "../data/usgs.sqlite3"
```

```
In [4]: records <- quickQuery db "SELECT * FROM earthquakes LIMIT 5" []
```

```

In [5]: records
[[SqlByteString "2016-07-11T19:44:16.470Z",SqlDouble 38.2513,SqlDouble (-115.9473),SqlDouble 20.2,SqlDouble 1.7],[SqlByteString "2016-07-11T19:38:52.140Z",SqlDouble 38.2467,SqlDouble (-118.6373),SqlDouble 7.0,SqlDouble 0.8],[SqlByteString "2016-07-11T19:27:19.750Z",SqlDouble 38.6712,SqlDouble (-117.0597),SqlDouble 0.3,SqlDouble 1.7],[SqlByteString "2016-07-11T19:01:47.820Z",SqlDouble 48.0008333,SqlDouble (-121.9131667),SqlDouble 11.92,SqlDouble 1.8],[SqlByteString "2016-07-11T18:48:12.000Z",SqlDouble 60.2407,SqlDouble (-141.6943),SqlDouble 13.7,SqlDouble 1.3]]

```

```

In [6]: length records
5

```

```
In [8]: readColumn = map fromSql
```

```

In [9]: readColumn (transpose records !! 3) :: [Double]
[20.2,7.0,0.3,11.92,13.7]

```

```
In [10]: northern <- quickQuery db "SELECT * FROM earthquakes WHERE latitude > 0" []
```

```
In [11]: length northern
```

```
1530
```

```
In [12]: southern <- quickQuery db "SELECT * FROM earthquakes WHERE latitude < 0" []
```

```
In [13]: length southern
```

```
55
```

```
In [14]: northwest <- quickQuery db "SELECT * FROM earthquakes WHERE latitude > 0 AND longitude < 0" []
```

```
In [15]: length northwest
```

```
1477
```

```
In [17]: southwest <- quickQuery db "SELECT * FROM earthquakes WHERE latitude < 0 AND longitude < 0" []
```

```
In [18]: length southwest
```

```
28
```

```
In [19]: readColumn (transpose southwest !! 3) :: [Double]
```

```
[8.12,45.25,271.75,35.56,114.67,10.0,10.0,10.0,10.0,10.0,177.68,118.76,35.41,36.89,10.0,10.0,109.76,10.0,36.59,44.27,94.22,36.89,10.0,10.0,10.0,10.0,39.51,11.2,93.55]
```

```
In [20]: readColumn (transpose southwest !! 4) :: [Double]
```

```
[4.7,4.9,4.7,4.3,4.8,4.6,5.8,6.0,5.0,4.7,4.4,4.6,5.3,5.1,5.1,4.2,4.6,4.4,4.3,4.5,4.8,5.2,4.9,5.9,5.2,4.3,5.0,4.7]
```

```
In [21]: southwest <- quickQuery db "SELECT * FROM earthquakes WHERE latitude < 0 AND longitude < 0 ORDER BY magnitude" []
```

```
In [22]: readColumn (transpose southwest !! 4) :: [Double]
```

```
[4.2,4.3,4.3,4.3,4.4,4.4,4.5,4.6,4.6,4.6,4.7,4.7,4.7,4.7,4.8,4.8,4.9,4.9,5.0,5.0,5.1,5.1,5.2,5.2,5.3,5.8,5.9,6.0]
```

```
In [23]: southwest <- quickQuery db "SELECT * FROM earthquakes WHERE latitude < 0 AND longitude < 0 ORDER BY magnitude LIMIT 5" []
```

```
In [24]: readColumn (transpose southwest !! 4) :: [Double]
```

```
[4.2,4.3,4.3,4.3,4.4]
```



```
In [25]: southwest <- quickQuery db "SELECT * FROM earthquakes WHERE latitude < 0 AND longitude < 0 ORDER BY magnitude DESC LIMIT 5"
```

```
In [26]: readColumn (transpose southwest !! 4) :: [Double]
[6.0,5.9,5.8,5.3,5.2]
```

```
jcchurch@dataanalysis:~/Code/HaskellDataAnalysis/analysis$ ls
Baseball.ipynb           Baseball-OpenCSV.ipynb   Earthquakes.ipynb
Baseball-Mean.ipynb     Baseball-Range.ipynb    Earthquakes-SELECT.ipynb
Baseball-Median.ipynb  DescriptiveStats.hs
Baseball-Mode.ipynb    Earthquakes-Connect.ipynb
jcchurch@dataanalysis:~/Code/HaskellDataAnalysis/analysis$
```

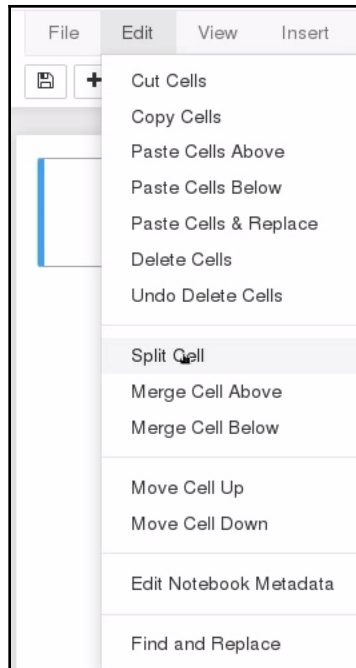
```
module DescriptiveStats where

import Data.List
import Data.Ord
import Data.Maybe

range :: Ord a => [a] -> Maybe (a, a)
range [] = Nothing
range [x] = Just (x, x)
range xs = Just (minimum xs, maximum xs)

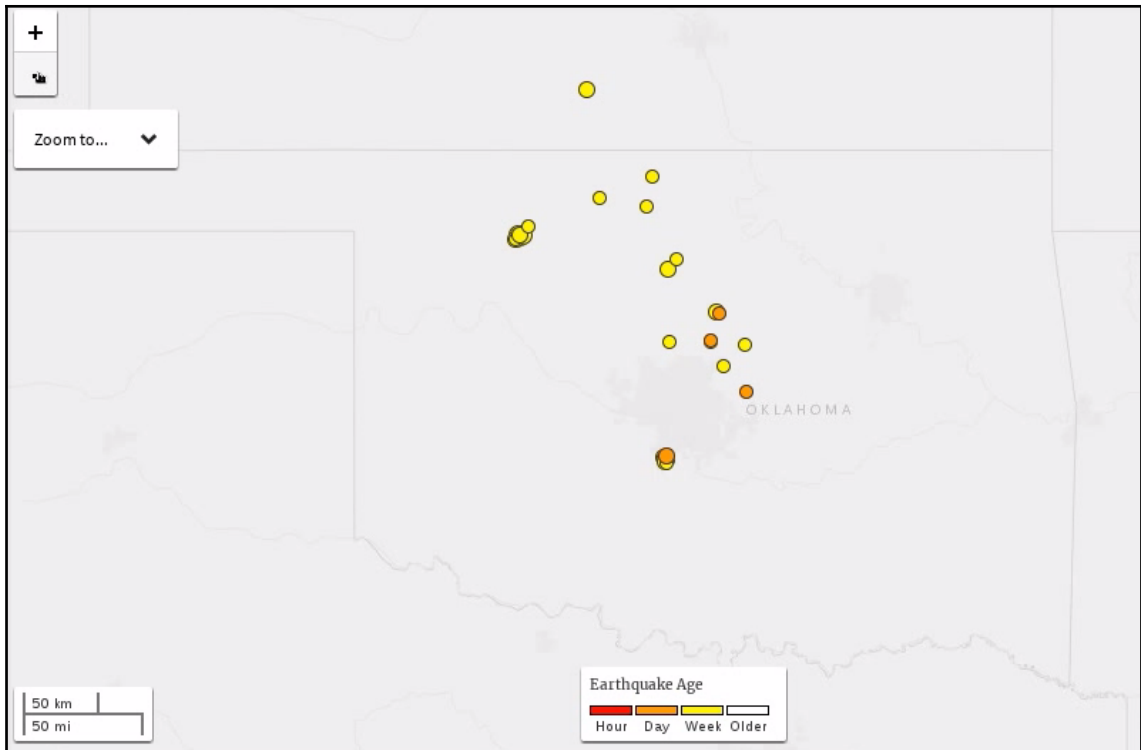
mean :: Real a => [a] -> Maybe Double
mean [] = Nothing
mean [x] = Just $ realToFrac x
mean xs = Just $ realToFrac (sum xs) / fromIntegral (length xs)

stdev :: Real a => [a] -> Maybe Double
stdev [] = Nothing
stdev [_] = Nothing
stdev xs = Just $ sqrt (sumsquares / n_m1)
  where
    n_m1 = fromIntegral (length xs - 1)
```



```
In [ ]: !load DescriptiveStats
```

```
In [2]: import Data.List
import Database.HDBC.SQLite3
import Database.HDBC
import DescriptiveStats
```



```
In [26]: length oklahoma
```

```
33
```

```
In [28]: length notoklahoma
```

```
1549
```

```
In [29]: okmag = readColumn (transpose oklahoma !! 4) :: [Double]
```

```
In [30]: notokmag = readColumn (transpose notoklahoma !! 4) :: [Double]
```

```
In [31]: mean okmag
Just 2.8424242424242423

In [32]: mean notokmag
Just 1.4760490639122006
```

```
In [33]: median okmag
Just 2.7

In [34]: median notokmag
Just 1.2
```

```
In [35]: stdev okmag
Just 0.5798870057122676

In [36]: stdev notokmag
Just 1.1934121006742073
```

```
In [37]: mode okmag
Just (2.5,7)

In [38]: mode notokmag
Just (1.2,47)
```

Chapter 3: Regular Expressions

```
jcchurch@dataanalysis:~/Code/HaskellDataAnalysis/analysis$ cabal install regex-posix
```

```
In [1]: import Text.Regex.Posix  
In [2]: str1 = "one fish two fish red fish blue fish"  
In [3]: str2 = "The quick brown fox jumps over the lazy dog."
```

```
In [4]: str1 =~ "one" :: Bool  
True
```

```
In [5]: str2 =~ "one" :: Bool  
False
```

```
In [6]: str1 =~ "o.e" :: Bool  
True  
In [7]: str2 =~ "o.e" :: Bool  
True
```

```
In [8]: str1 =~ "fish|fox" :: Bool  
True  
In [9]: str2 =~ "fish|fox" :: Bool  
True
```

```
In [10]: str1 =~ "dog|cat" :: Bool  
False  
In [11]: str1 =~ "dog|cat" :: Bool  
False
```

```
In [43]: "1969-07-20" =~ "(19|20)....." :: Bool
True
```

```
In [45]: "18AB-40-99" =~ "(19|20)....." :: Bool
False
```

Spellings:

- color
- colour
- Wrong: coluor

```
In [46]: "color" =~ "colou?r" :: Bool
True
```

```
In [47]: "colour" =~ "colou?r" :: Bool
True
```

```
In [48]: "coluor" =~ "colou?r" :: Bool
False
```

```
In [49]: "1122" =~ "1*2*" :: Bool
True
```

```
In [50]: "" =~ "1*2*" :: Bool
True
```

```
In [51]: "111" =~ "1*2*" :: Bool
True
```

```
In [52]: "ABC" =~ "1*2*" :: Bool
```

```
True
```

```
In [53]: "1122" =~ "1+2+" :: Bool
```

```
True
```

```
In [54]: "11" =~ "1+2+" :: Bool
```

```
False
```

```
In [55]: "12" =~ "1+2+" :: Bool
```

```
True
```

```
In [56]: "123" =~ "12{3,5}3" :: Bool
```

```
False
```

```
In [57]: "1223" =~ "12{3,5}3" :: Bool
```

```
False
```

```
In [58]: "12223" =~ "12{3,5}3" :: Bool
```

```
True
```

```
In [59]: "12222223" =~ "12{3,5}3" :: Bool
```

```
False
```

```
In [36]: "dog" =~ "[aeiou]" :: Bool
```

```
True
```

```
In [37]: "why" =~ "[aeiou]" :: Bool
```

```
False
```

```
In [38]: "123" =~ "[0123456789]" :: Bool
```

```
True
```

```
In [39]: "dog" =~ "[0123456789]" :: Bool
False
```

```
In [40]: "123" =~ "[0-9]" :: Bool
True
```

```
In [41]: "dog" =~ "[a-z]" :: Bool
True
```

```
In [42]: "dog" =~ "[A-Z]" :: Bool
False
```

```
In [43]: "DoG" =~ "[A-Za-z]" :: Bool
True
```

```
In [44]: "why" =~ "^[aeiou]" :: Bool
True
```

```
In [45]: "Ke$ha" =~ "^[A-Za-z]" :: Bool
True
```

```
In [46]: date1 = "1969-07-20"
```

```
In [47]: date1 =~ "(19|20)[0-9][0-9]-[0-9][0-9]-[0-9][0-9]" :: Bool
True
```

```
In [48]: "1901-40-99" =~ "(19|20)[0-9][0-9]-[0-9][0-9]-[0-9][0-9]" :: Bool
True
```



```
In [3]: baseball <- parseCSVFromFile "../data/GL2015.TXT"
```

```
In [4]: gameDates = getIndex baseball 0
```

```
In [5]: head gameDates
```

```
"20150405"
```

```
In [6]: marchDates = map (==~ "...04..") gameDates :: [Bool]
```

```
In [7]: awayRuns = readIndex baseball 9 :: [Integer]
```

```
In [8]: length marchDates
```

```
2429
```

```
In [9]: length awayRuns
```

```
2429
```

```
In [10]: zip marchDates awayRuns
```

```
[(True,3), (True,0), (True,0), (True,1), (True,6), (True,0), (True,1), (True,6), (True,5), (True,2), (True,3), (True,2), (True,10), (True,8), (True,3), (True,3), (True,2), (True,6), (True,6), (True,7), (True,12), (True,5), (True,0), (True,2), (True,5), (True,3), (True,0), (True,5), (True,0), (True,5), (True,0), (True,4), (True,4), (True,2), (True,5), (True,2), (True,1), (True,1), (True,5), (True,1), (True,6), (True,10), (True,2), (True,6), (True,1), (True,6), (True,4), (True,12), (True,6), (True,8), (True,6), (True,0), (True,5), (True,3), (True,3), (True,4), (True,1), (True,9), (True,6), (True,1), (True,0), (True,6), (True,1), (True,4), (True,9), (True,8), (True,5), (True,2), (True,0), (True,3), (True,4), (True,9), (True,2), (True,0), (True,2), (True,2), (True,9), (True,10), (True,2), (True,8), (True,4), (True,8), (True,6), (True,7), (True,4), (True,7), (True,6), (True,8), (True,10), (True,4), (True,4), (True,6), (True,4), (True,8), (True,8), (True,12), (True,6), (True,2), (True,2), (True,6), (True,5), (True,0), (True,4), (True,8), (True,2), (True,5), (True,3), (True,7), (True,4), (True,4), (True,2), (True,3), (True,8), (True,3), (True,5), (True,5), (True,2), (True,1), (True,4), (True,5), (True,10), (True,2), (True,1), (True,1), (True,10), (True,7), (True,6), (True,0), (True,2), (True,1), (True,1), (True,2), (True,4), (True,2), (True,5), (True,4), (True,5), (True,7), (True,0), (True,2), (True,2), (True,1), (True,6), (True,4), (True,2), (True,3), (True,5), (True,8), (True,5), (True,3), (True,1), (True,3), (True,9), (True,1), (True,2), (True,4), (True,12), (True,0), (True,5), (True,4), (True,1), (True,9), (True,5), (True,6), (True,3), (True,4), (True,2), (True,1), (True,2), (True,5), (True,8), (True,1), (True,3), (True,2), (True,2), (True,10), (True,5), (True,5), (True,5), (True,0), (True,6), (True,2), (True,5), (True,1), (True,1), (True,6), (True,1), (True,3)
```



```
In [8]: printf "%02d" 7  
T07
```

```
In [9]: printf "%02d" 19  
T19
```

```
In [10]: countAtHour :: Int -> Int  
countAtHour hour = length $ filter (== (printf "%02d" hour :: String)) timestamps
```

```
In [11]: countAtHour 0  
76
```

```
In [12]: countAtHour 1  
55
```

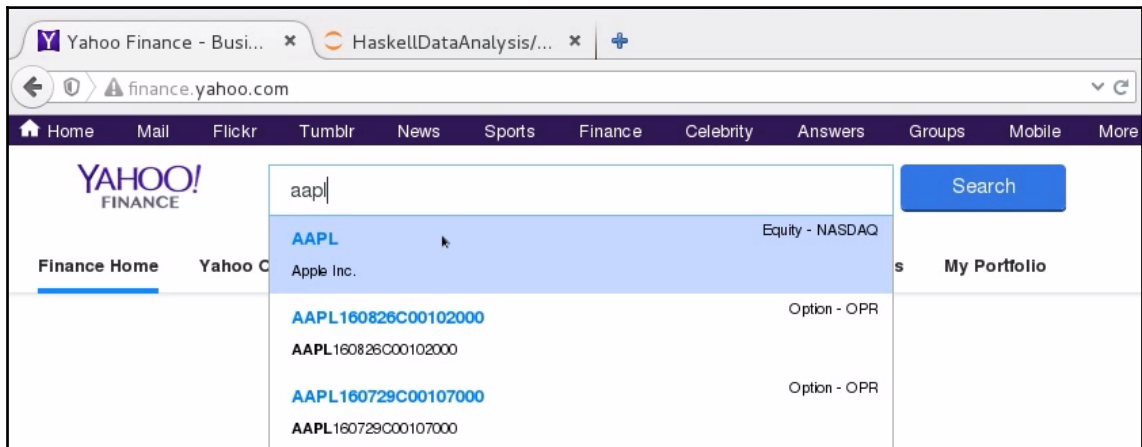
```
In [13]: countAtHour 2  
52
```

```
In [14]: map countAtHour [0..23]  
[76,55,52,65,68,65,77,78,59,75,58,70,70,68,61,72,57,62,64,64,57,64,64,84]
```

Chapter 4: Visualizations

```
jcchurch@dataanalysis:~/Downloads$ sudo apt-get install gnuplot
```

```
jcchurch@dataanalysis:~/Downloads$ cabal update
Downloading the latest package list from hackage.haskell.org
jcchurch@dataanalysis:~/Downloads$ cabal install easyplot
Resolving dependencies...
cabal: Entering directory '/tmp/cabal-tmp-6405/easyplot-1.0'
Configuring easyplot-1.0...
Building easyplot-1.0...
Preprocessing library easyplot-1.0...
[1 of 1] Compiling Graphics.EasyPlot ( src/Graphics/EasyPlot.hs, dist/build/Graph
ics/EasyPlot.o )
Creating package registration file: /tmp/pkgConf-easyplot-16405.0
Installing library in
/home/jcchurch/.cabal/lib/x86_64-linux-ghc-7.6.3/easyplot-1.0-CcbZmtpo0waLPtHw9r
xdq8
Registering easyplot-1.0...
cabal: Leaving directory '/tmp/cabal-tmp-6405/easyplot-1.0'
Installed easyplot-1.0
```



The screenshot shows a web browser window with two tabs: 'Yahoo Finance - Busi...' and 'HaskellDataAnalysis/...'. The address bar shows 'finance.yahoo.com'. The navigation bar includes links for Home, Mail, Flickr, Tumblr, News, Sports, Finance, Celebrity, Answers, Groups, Mobile, and More. The main content area features the Yahoo! Finance logo and a search bar with 'aapl' entered. A blue 'Search' button is to the right. Below the search bar, a table of search results is displayed:

Symbol	Company Name	Market
AAPL	Apple Inc.	Equity - NASDAQ
AAPL160826C00102000		Option - OPR
AAPL160729C00107000		Option - OPR

Additional navigation links include 'Finance Home', 'Yahoo C', and 'My Portfolio'.

Summary Conversations Statistics Profile Financials Options Holders **Historical Data** Analysts

Time Period: Jul 19, 2015 - Jul 19, 2016 Show: Historical Prices Frequency: Daily Apply

Currency in USD. [Download Data](#)

Date	Open	High	Low	Adj Close*	Volume
Jul 19, 2016	99.50	100.00	99.34	99.65	11,562,883
Jul 18, 2016	98.70	100.13	98.60	99.83	36,439,900

Time Period: Jul 19, 2015 - Jul 19, 2016 Show: Historical Prices Frequency: Daily Apply

Currency in USD. [Download Data](#)

Date	Open	High	Low	Adj Close*	Volume
Jul 19, 2016	99.50	100.00	99.34	99.65	11,562,883
Jul 18, 2016	98.70	100.13	98.60	99.83	36,439,900
Jul 15, 2016	98.92	99.30	98.50	98.78	30,137,000
Jul 14, 2016	97.39	98.99	97.32	98.79	38,919,000
Jul 13, 2016	97.41	97.67	96.84	96.87	25,892,200

Opening table.csv

You have chosen to open:

- table.csv
which is: CSV document
from: http://chart.finance.yahoo.com

What should Iceweasel do with this file?

- Open with gedit (default)
- Save File
- Do this automatically for files like this from now on.

Cancel OK

Time Period: Jan 01, 1960 - Jul 19, 2016

Currency in USD. [Download Data](#)

Date	Open	High	Low	Adj Close*	Volume
Jul 18, 2016	98.70	100.13	98.60	99.83	36,439,900

```
jcchurch@dataanalysis:~/Downloads$ ls  
all_week.csv  earthquakes.csv  GL2015.TXT  gl2015.zip  table.csv  usgs.sqlite3  
jcchurch@dataanalysis:~/Downloads$ mv table.csv aapl.csv  
jcchurch@dataanalysis:~/Downloads$ vi aapl.csv
```

```

Date,Open,High,Low,Close,Volume,Adj Close
2016-07-18,98.699997,100.129997,98.599998,99.830002,36439900,99.830002
2016-07-15,98.919998,99.300003,98.50,98.779999,29952500,98.779999
2016-07-14,97.389999,98.989998,97.32,98.790001,38348800,98.790001
2016-07-13,97.410004,97.669998,96.839996,96.870003,25655000,96.870003
2016-07-12,97.169998,97.699997,97.120003,97.419998,23889600,97.419998
2016-07-11,96.75,97.650002,96.730003,96.980003,23298900,96.980003
2016-07-08,96.489998,96.889999,96.050003,96.68,28855800,96.68
2016-07-07,95.699997,96.50,95.620003,95.940002,24280900,95.940002
2016-07-06,94.599998,95.660004,94.370003,95.529999,30770700,95.529999
2016-07-05,95.389999,95.400002,94.459999,95.040001,27257000,95.040001
2016-07-01,95.489998,96.470001,95.330002,95.889999,25872300,95.889999
2016-06-30,94.440002,95.769997,94.300003,95.599998,35119400,95.599998
2016-06-29,93.970001,94.550003,93.629997,94.400002,36427800,94.400002
2016-06-28,92.900002,93.660004,92.139999,93.589996,39311500,93.589996
2016-06-27,93.00,93.050003,91.50,92.040001,45489600,92.040001
2016-06-24,92.910004,94.660004,92.650002,93.400002,72894000,93.400002
2016-06-23,95.940002,96.290001,95.25,96.099998,31863500,96.099998
2016-06-22,96.25,96.889999,95.349998,95.550003,28971100,95.550003
2016-06-21,94.940002,96.349998,94.68,95.910004,35229500,95.910004
2016-06-20,96.00,96.57,95.029999,95.099998,33942300,95.099998
2016-06-17,96.620003,96.650002,95.300003,95.330002,60595000,95.330002
"apl.csv" 8977 lines, 582517 characters

```

```

1980-12-23,30.874999,30.999999,30.874999,30.874999,11737600,0.463241
1980-12-22,29.625,29.75,29.625,29.625,9340800,0.444486
1980-12-19,28.249999,28.374999,28.249999,28.249999,12157600,0.423856
1980-12-18,26.625,26.75,26.625,26.625,18362400,0.399475
1980-12-17,25.875,25.999999,25.875,25.875,21610400,0.388222
1980-12-16,25.375,25.375,25.25,25.25,26432000,0.378845
1980-12-15,27.375001,27.375001,27.25,27.25,43971200,0.408852
1980-12-12,28.75,28.875,28.75,28.75,117258400,0.431358

```

Files
Running
Clusters

Select items to perform actions on them.

/ HaskellDataAnalysis / data

	..
	GL2015.TXT
	stocks.sqlite3
	usgs.sqlite3


```
In [1]: import Graphics.EasyPlot
import Database.HDBC
import Database.HDBC.SQLite3
```

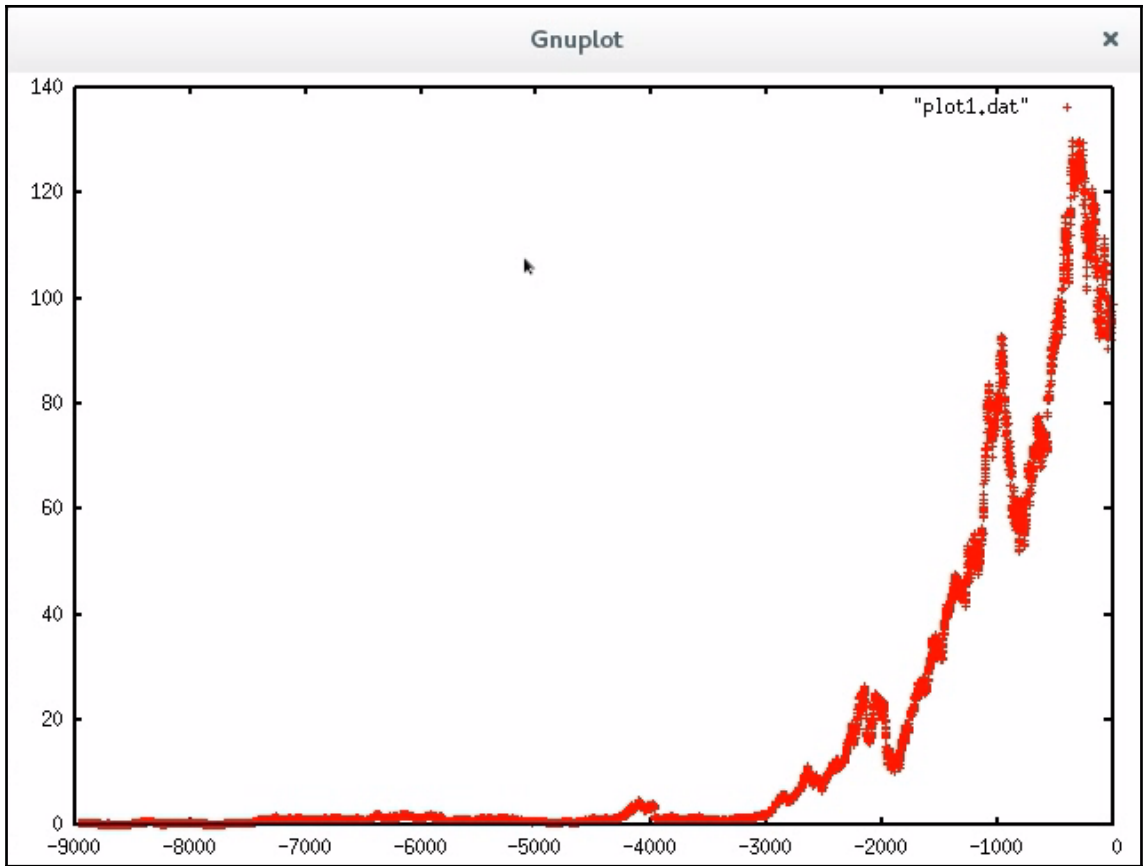
```
In [2]: db <- connectSQLite3 "../data/stocks.sqlite3"
```

```
In [3]: aaplRaw <- quickQuery db "SELECT adjclose FROM aapl" []
```

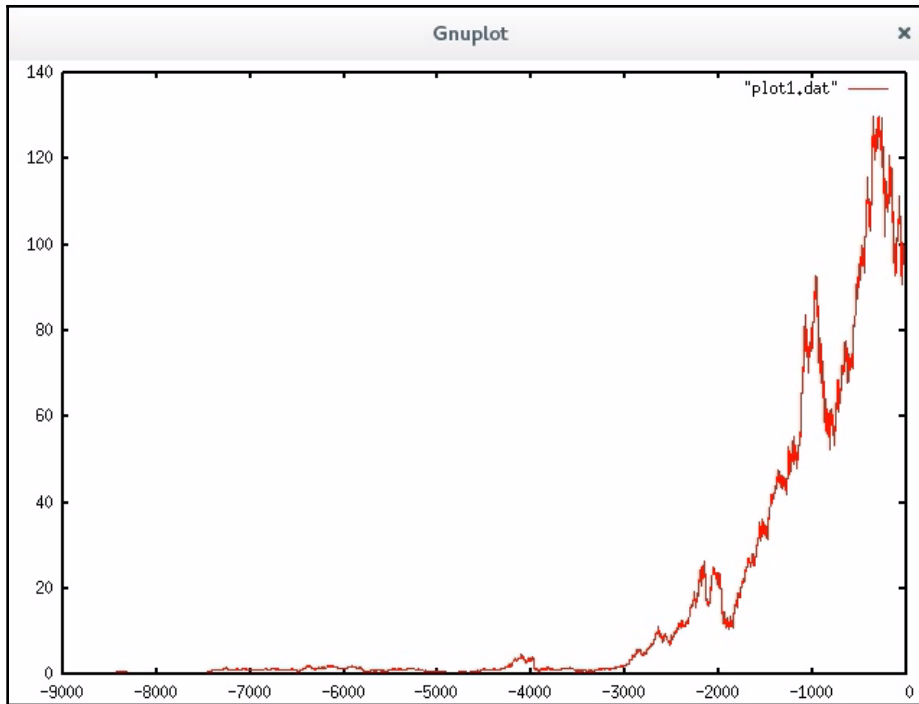
```
In [4]: :t aaplRaw
aaplRaw :: [[SqlValue]]
```

```
In [5]: aapl = map (fromSql . head) aaplRaw :: [Double]
```

```
In [12]: plot X11 $ zip [0,-1..] aapl
True
```



```
In [13]: plot X11 $ Data2D [Style Lines] [] (zip [0,-1..] aapl)
True
```



```
In [14]: movingSum :: [Double] -> [Double] -> [Double]
movingSum xs [] = [sum xs]
movingSum (x:xs) (y:ys) = (sum (x:xs)):movingSum (xs++[y]) ys
```

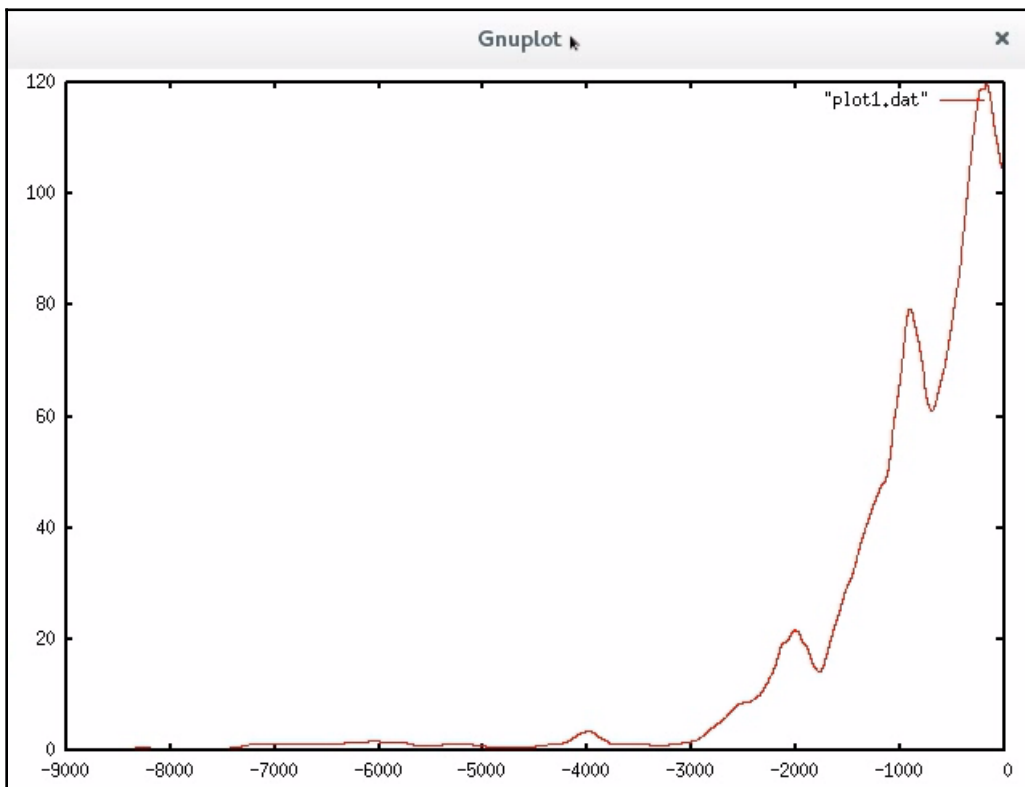
```
In [15]: movingSum [1,2,3] [4,5,6,7,8,9,10]
[6.0,9.0,12.0,15.0,18.0,21.0,24.0,27.0]
```

```
In [16]: movingAverage :: [Double] -> Int -> [Double]
movingAverage xs windowSize
| windowSize < 1 = error "Cannot have non-positive window size."
| windowSize > length xs = []
| otherwise = map (/ fromIntegral windowSize) movingSums
  where
    movingSums = movingSum (take windowSize xs) (drop windowSize xs)
```

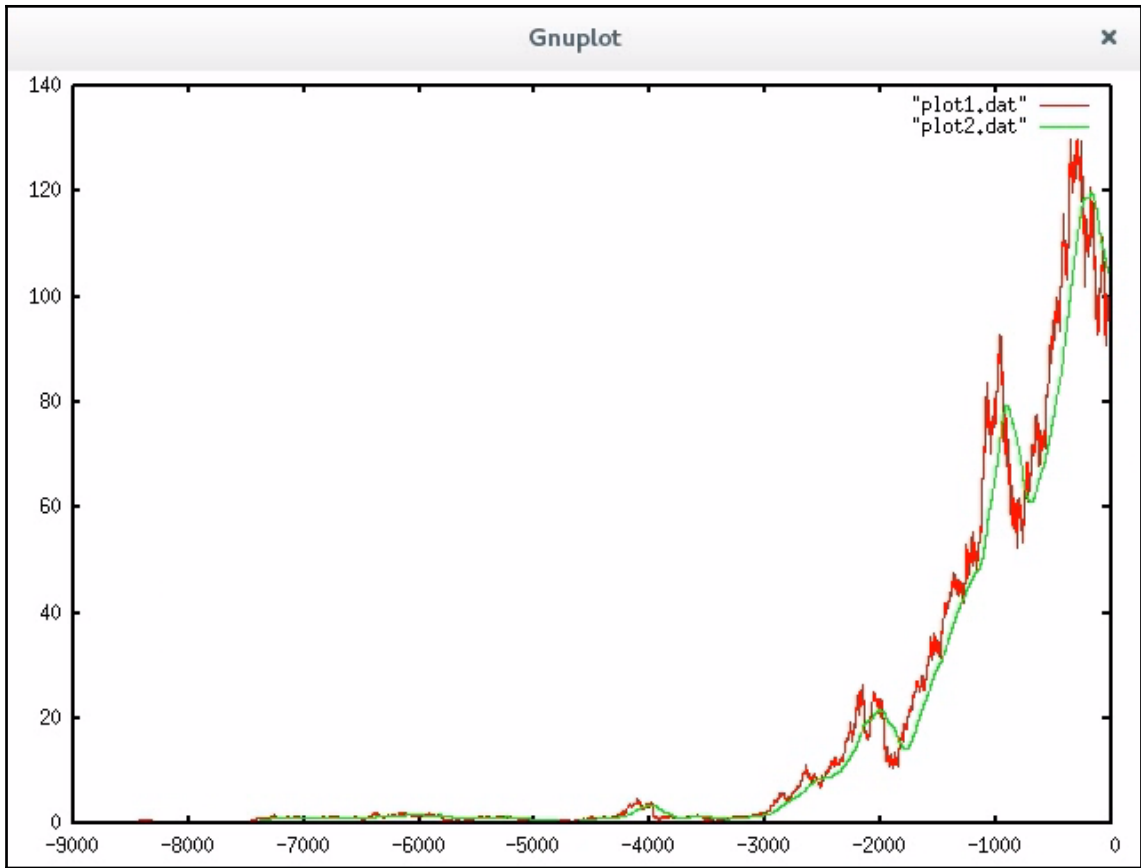
```
In [17]: movingAverage [1..10] 3
         [2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0]
```

```
In [18]: aaplMA = movingAverage aapl 200
```

```
In [21]: plot X11 $ Data2D [Style Lines] [] (zip [0,-1..] aaplMA)
         True
```

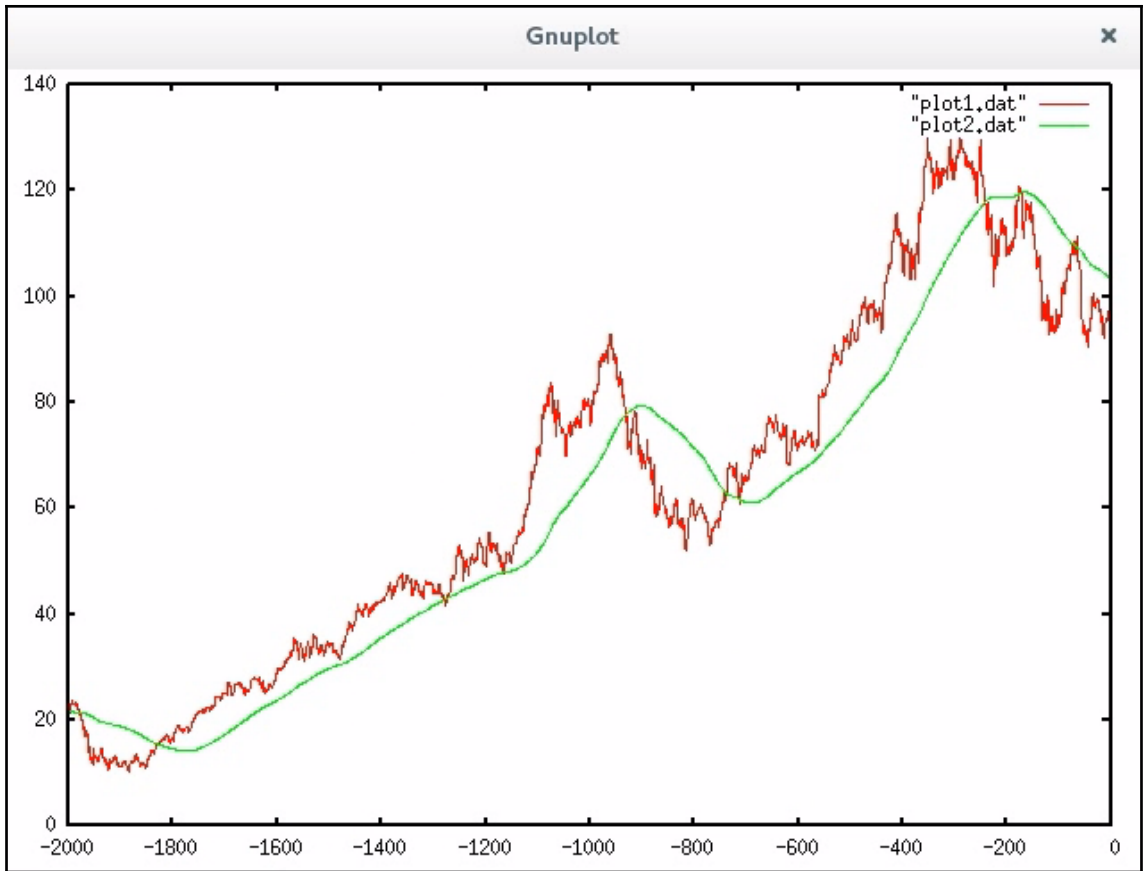


```
In [22]: plot X11 [Data2D [Style Lines] [] (zip [0,-1..] aapl), Data2D [Style Lines] [] (zip [0,-1..] aaplMA)]
         True
```

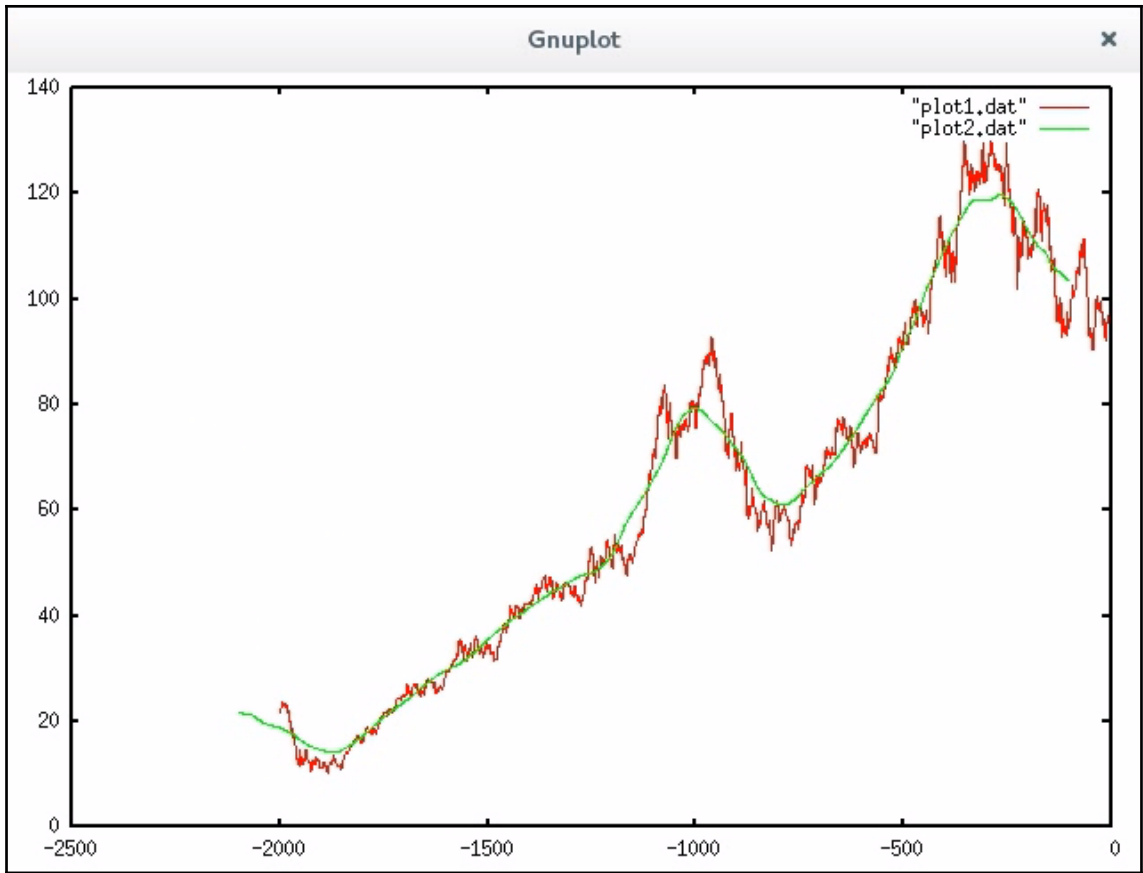


```
In [23]: plot X11  
         [ Data2D [Style Lines] [] (zip [0,-1..] (take 2000 aapl))  
         , Data2D [Style Lines] [] (zip [0,-1..] (take 2000 aaplMA))  
         ]
```

True

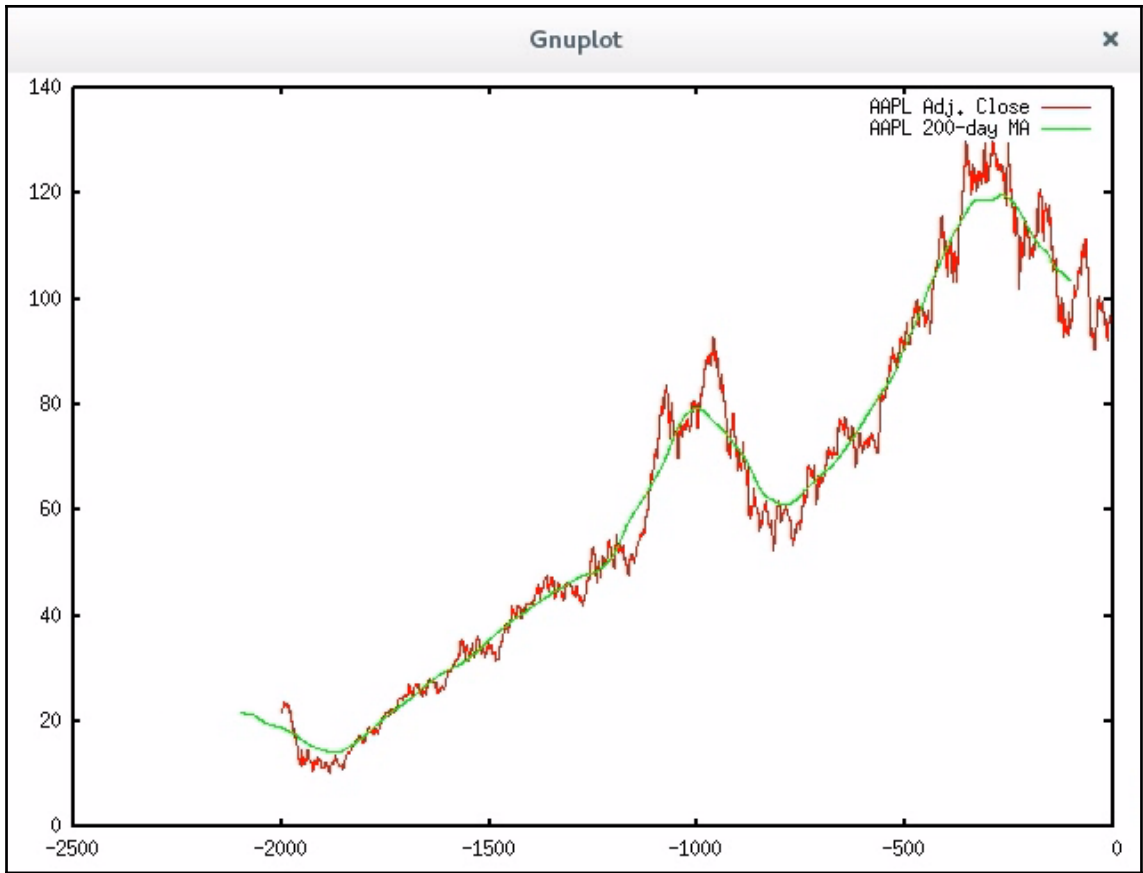


```
In [*]: plot X11
        [ Data2D [Style Lines] [] (zip [0,-1..] (take 2000 aapl))
          , Data2D [Style Lines] [] (zip [-100,-101..] (take 2000 aaplMA))
        ]
True
```

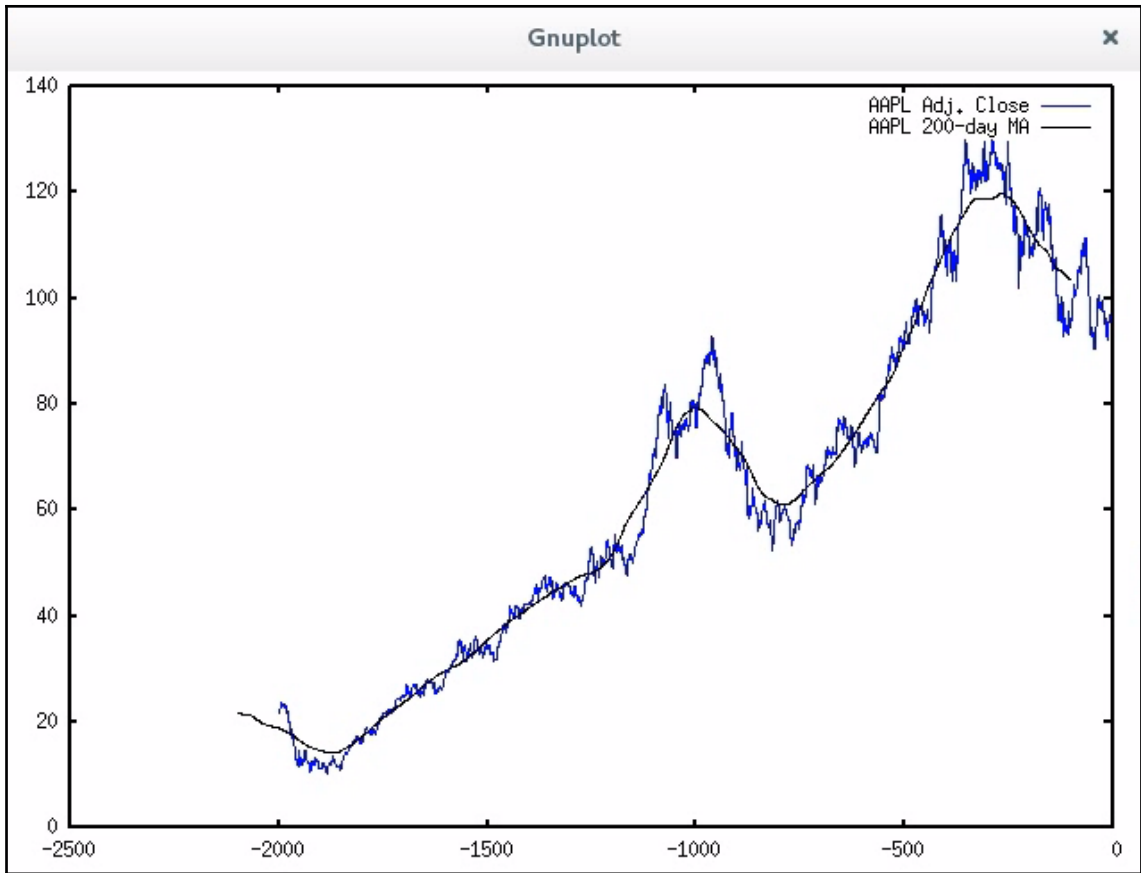


```
In [25]: plot X11
         [ Data2D [Style Lines, Title "AAPL Adj. Close"] [] (zip [0,-1..] (take 2000 aapl))
         , Data2D [Style Lines, Title "AAPL 200-day MA"] [] (zip [-100,-101..] (take 2000 aaplMA))
         ]
```

True

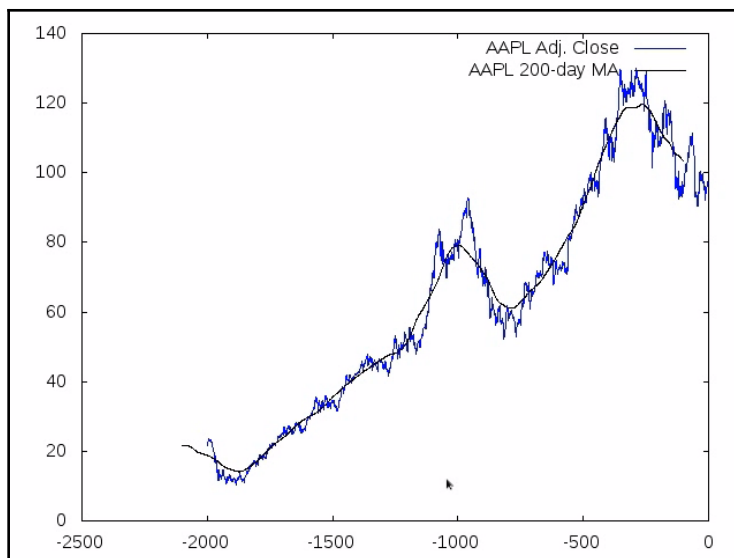


```
In [26]: plot X11
[ Data2D [Style Lines, Title "AAPL Adj. Close", Color Blue] [] (zip [0,-1..] (take 2000 aapl))
, Data2D [Style Lines, Title "AAPL 200-day MA", Color Black] [] (zip [-100,-101..] (take 2000 aaplMA))
]
True
```

```
In [27]: plot (PNG "adjcloseWith200DayMA.png")
         [ Data2D [Style Lines, Title "AAPL Adj. Close", Color Blue] [] (zip [0,-1..] (take 2000 aapl))
         , Data2D [Style Lines, Title "AAPL 200-day MA", Color Black] [] (zip [-100,-101..] (take 2000 aaplMA))
         ]
True
```

```
jcchurch@dataanalysis:~/Code/HaskellDataAnalysis/analysis$ ls
adjcloseWith200DayMA.png  Earthquakes-Connect.ipynb
Baseball.ipynb           Earthquakes.ipynb
Baseball-Mean.ipynb      Earthquakes-SELECT.ipynb
Baseball-Median.ipynb   plot1.dat
Baseball-Mode.ipynb     plot2.dat
Baseball-OpenCSV.ipynb  StockAnalysis-FirstPlot.ipynb
Baseball-Range.ipynb    StockAnalysis.ipynb
DescriptiveStats.hs      StockAnalysis-MovingAverage.ipynb
jcchurch@dataanalysis:~/Code/HaskellDataAnalysis/analysis$ iceweasel adjcloseWith200DayMA.png
```



$$\frac{x - X_1}{\max(X) - \min(X)}$$

In [36]:

```
googlRaw <- quickQuery db "SELECT adjclose FROM googl" []
msftRaw <- quickQuery db "SELECT adjclose FROM msft" []
googl = map (fromSql . head) googlRaw :: [Double]
msft = map (fromSql . head) msftRaw :: [Double]
```

In [37]: aapl252 = take 252 aapl

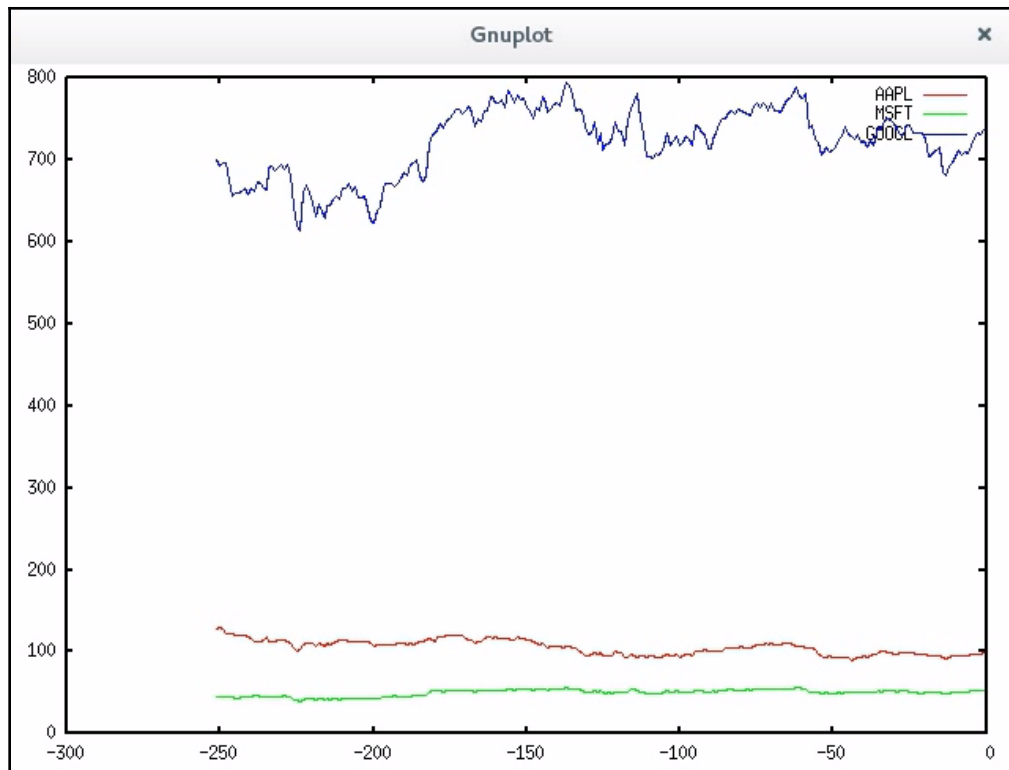
In [38]: msft252 = take 252 msft

In [39]: googl252 = take 252 googl

In [40]:

```
plot X11
 [ Data2D [Style Lines, Title "AAPL"] [] (zip [0,-1..] aapl252)
 , Data2D [Style Lines, Title "MSFT"] [] (zip [0,-1..] msft252)
 , Data2D [Style Lines, Title "GOOGL"] [] (zip [0,-1..] googl252)
 ]
```

True



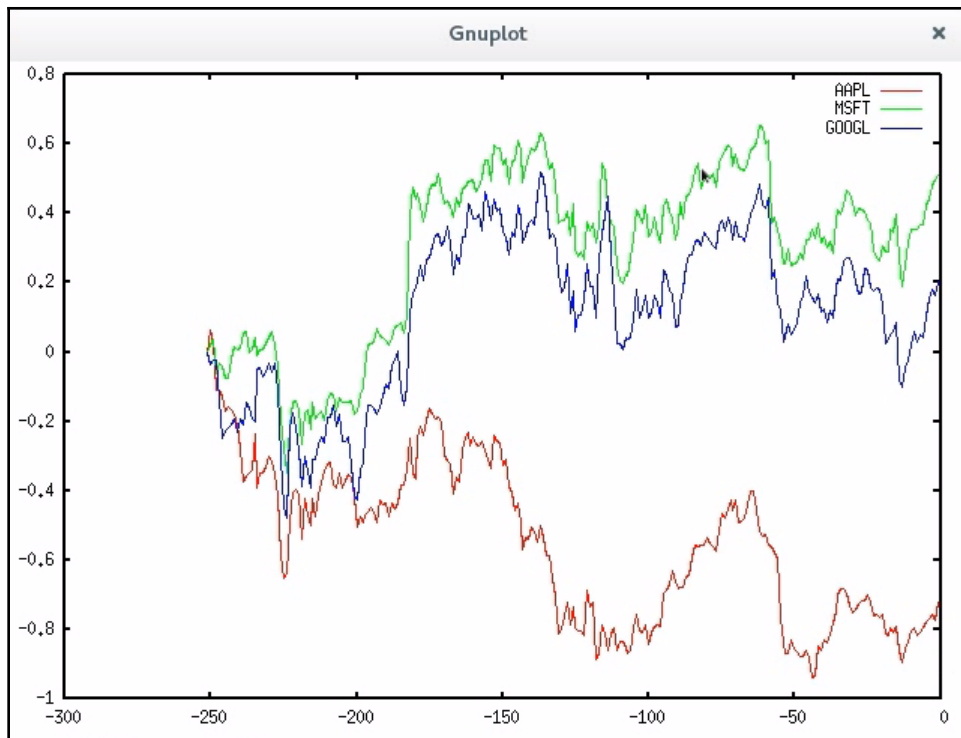
In [41]:

```
rescale :: [Double] -> [Double]
rescale xs = map (\x -> (x - lead)/(diffxs)) xs
  where
    lead = last xs
    maxx = maximum xs
    minx = minimum xs
    diffxs = maxx - minx
```

In [26]:

```
plot X11
[ Data2D [Style Lines, Title "AAPL"] [] (zip [0,-1..] (rescale aapl252))
, Data2D [Style Lines, Title "MSFT"] [] (zip [0,-1..] (rescale msft252))
, Data2D [Style Lines, Title "GOOGL"] [] (zip [0,-1..] (rescale googl252))
]
```

True

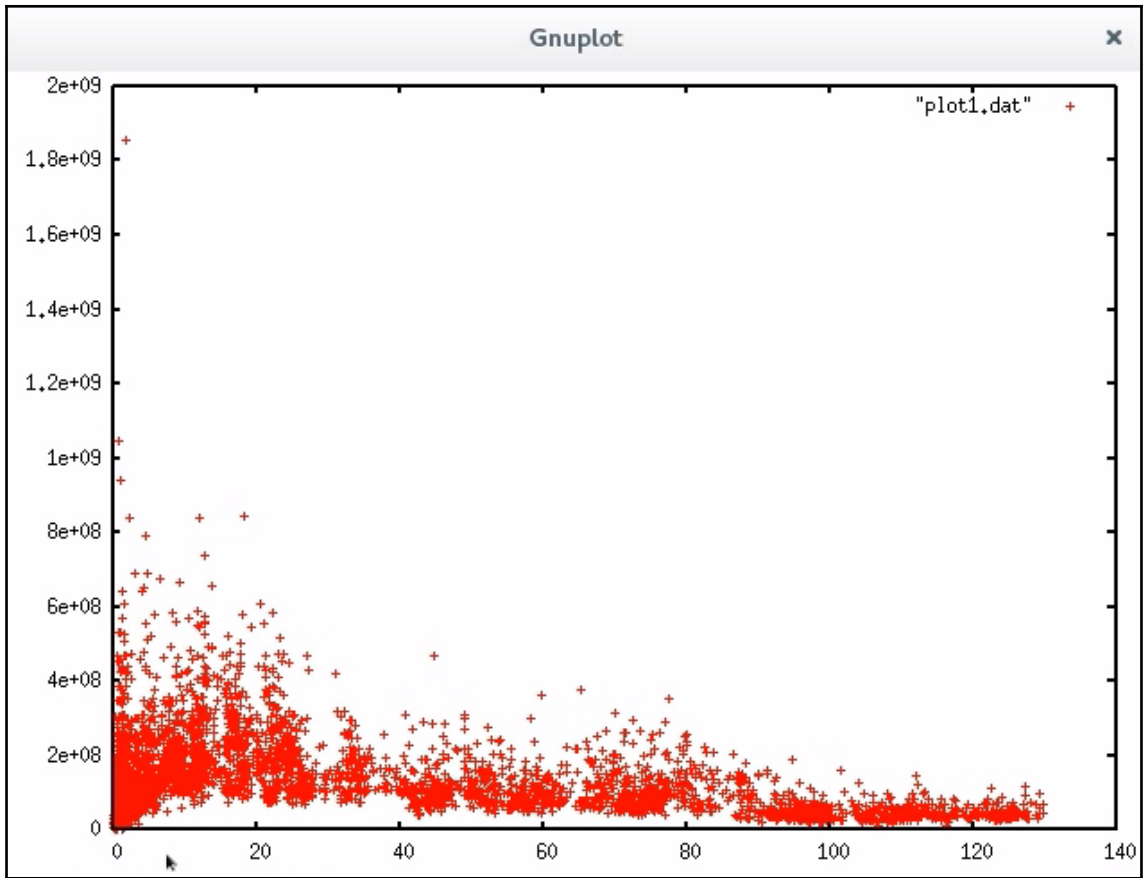


In [27]:

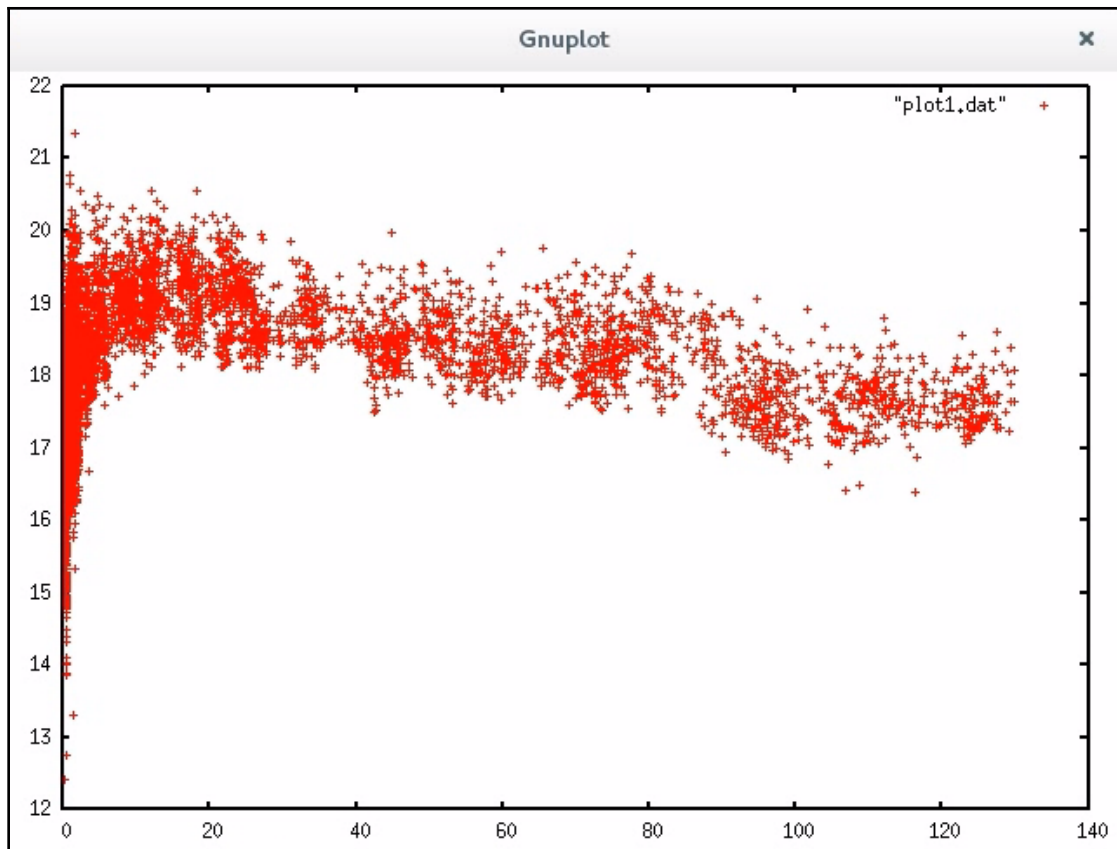
```
aaplVolRaw <- quickQuery db "SELECT volume FROM aapl" []  
msftVolRaw <- quickQuery db "SELECT volume FROM msft" []  
googlVolRaw <- quickQuery db "SELECT volume FROM googl" []  
aaplVol = map (fromSql . head) aaplVolRaw :: [Double]  
msftVol = map (fromSql . head) msftVolRaw :: [Double]  
googlVol = map (fromSql . head) googlVolRaw :: [Double]
```

In [28]: `plot X11 $ zip aapl aaplVol`

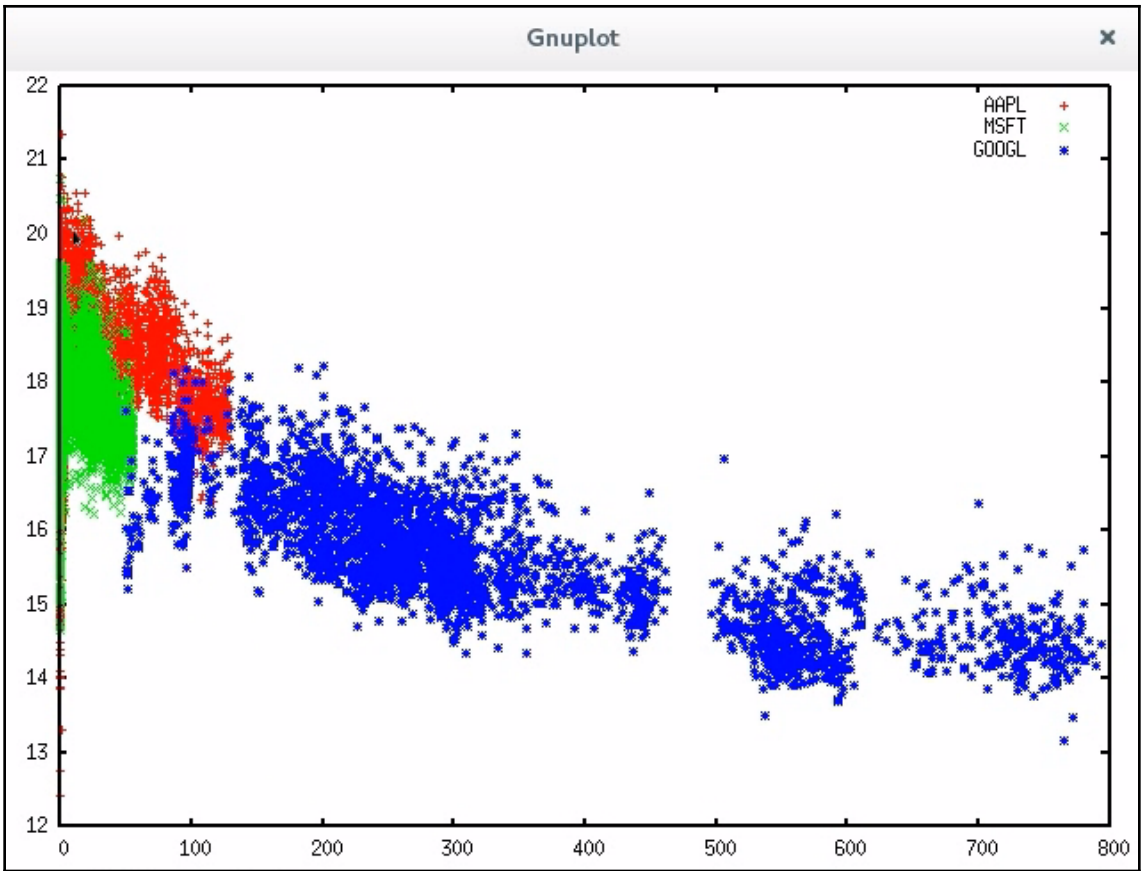
True



```
In [29]: plot X11 $ zip aapl (map log aaplVol)
True
```



```
In [*]: plot X11 [ Data2D [Title "AAPL"] [] (zip aapl (map log aaplVol))  
              , Data2D [Title "MSFT"] [] (zip msft (map log msftVol))  
              ; Data2D [Title "GOOGL"] [] (zip googl (map log googlVol))  
              ]
```



Chapter 5: Kernel Density Estimation

```
In [1]: !load DescriptiveStats
```

```
In [2]: import System.Random
import Data.List
import Graphics.EasyPlot
import DescriptiveStats
```

```
In [3]: g <- newStdGen
```

```
In [5]: values = take 100000 (randomRs (1,100) g) :: [Double]
```

```
In [6]: take 10 values
```

```
[58.78013806951732, 3.050543072227301, 29.024389411225123, 85.73139042381948, 69.36536178205485, 99.0783590607034, 58.17348331506121, 87.13442820846359, 54.61911359384984, 25.05354227810114]
```

```
In [7]: chunk [] = []
chunk list = take 10 list ; chunk (drop 10 list)
```

```
In [8]: chunkedValues = chunk values
```

```
In [9]: length chunkedValues
```

```
10000
```

```
In [10]: avgs = map (\xs -> sum xs / genericLength xs) chunkedValues
```

```
In [11]: pairs = runLengthEncoding $ sort (map round avgs)
```



```
In [12]: plot X11 $ Data2D [Style Lines] [] (map (\(x,y) -> (fromIntegral x, y)) pairs)
```

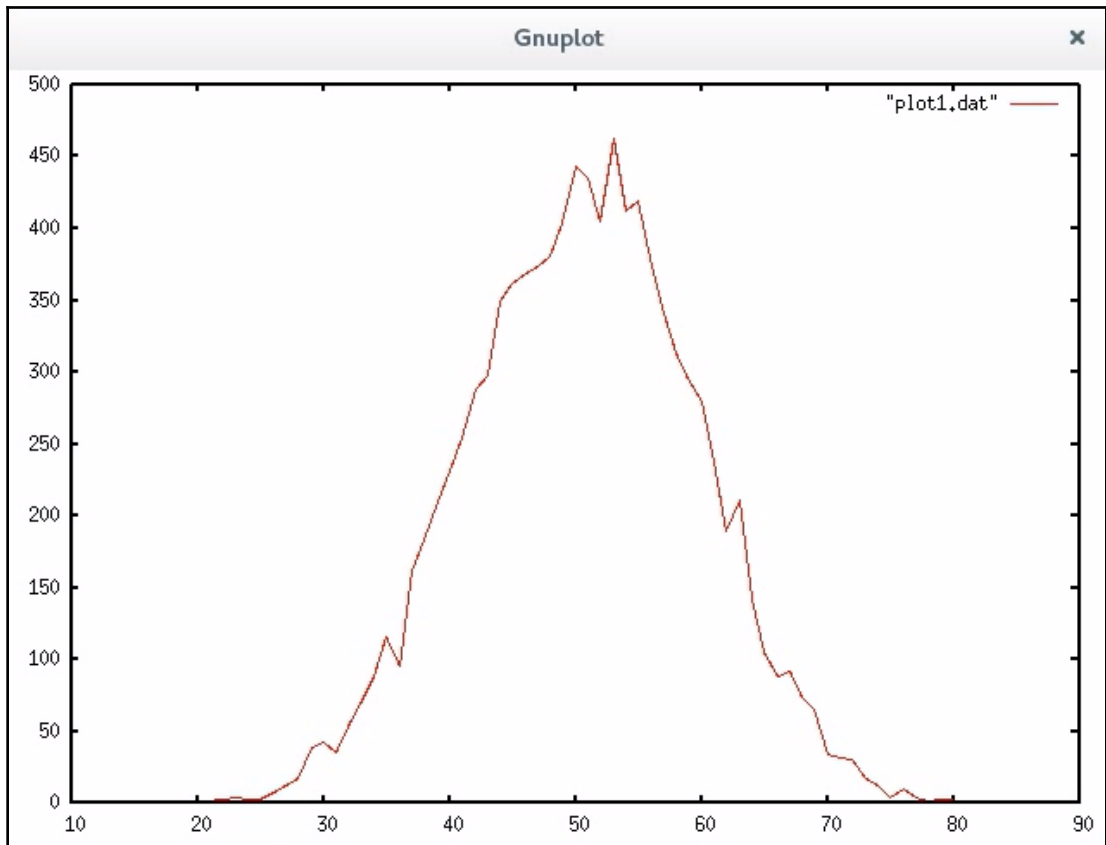
Use first

Found:

Why Not:

```
\ (x, y) -> (fromIntegral x, y) Control.Arrow.first fromIntegral
```

True

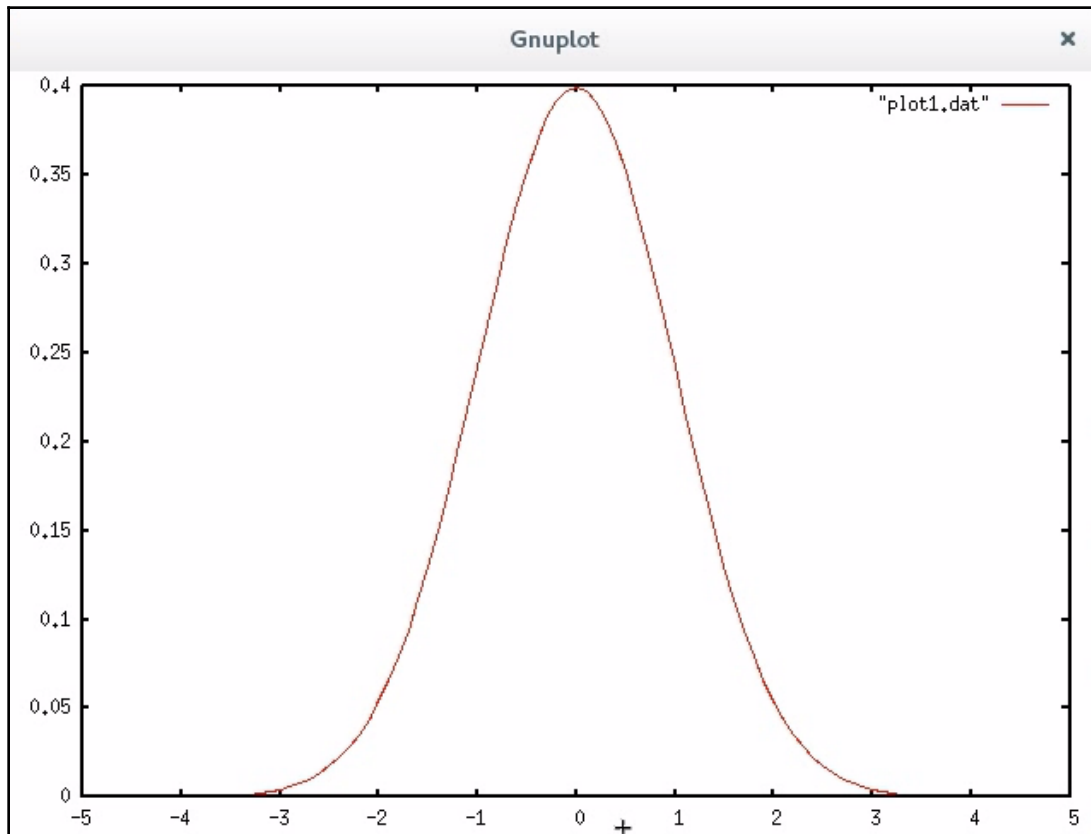


$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

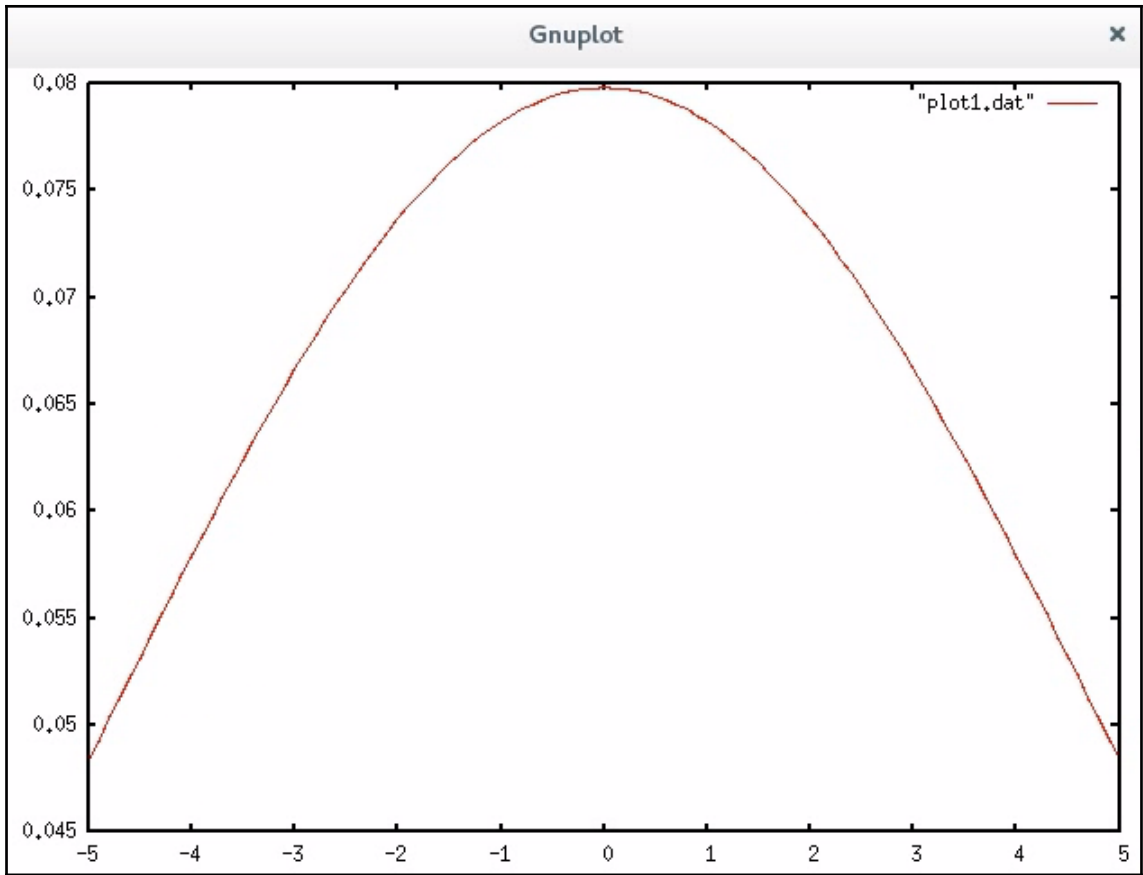
```
In [13]: normal mu sd x = exp (-((x-mu)^2 / (2*sd*sd))) / sqrt (2*sd*sd*pi)
```

```
In [14]: domain = [-5,-4.9..5]
```

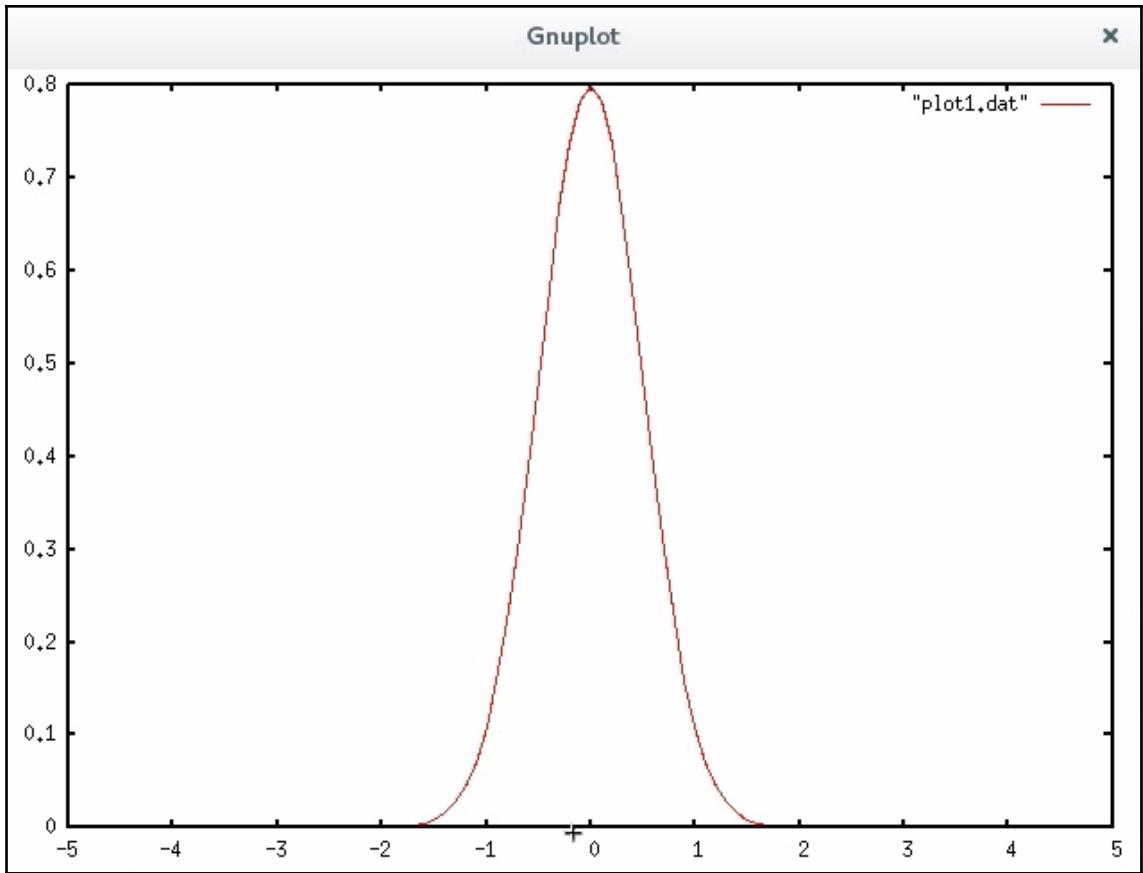
```
In [15]: plot X11 $ Data2D [Style Lines] [] (zip domain (map (normal 0 1) domain))  
True
```



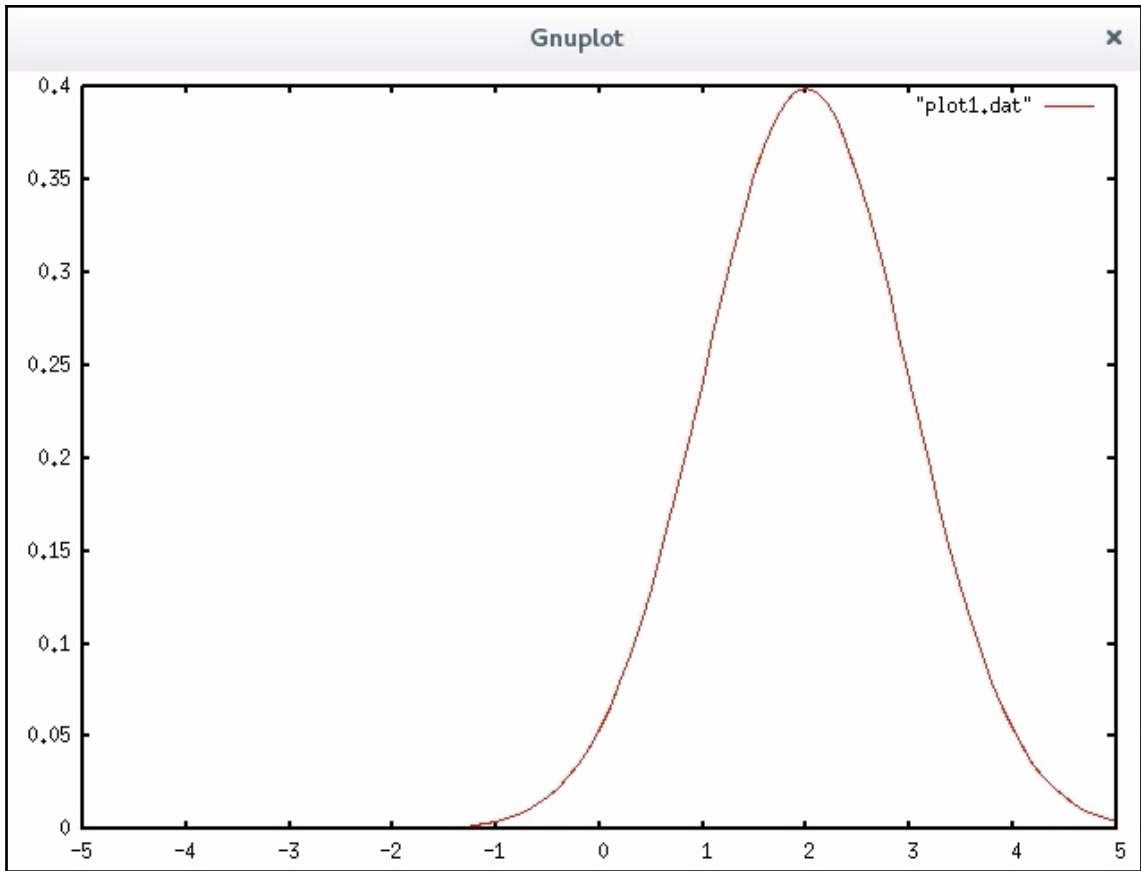
```
In [16]: plot X11 $ Data2D [Style Lines] [] (zip domain (map (normal 0 5) domain))  
True
```



```
In [17]: plot X11 $ Data2D [Style Lines] [] (zip domain (map (normal 0 0.5) domain))
True
```



```
In [18]: plot X11 $ Data2D [Style Lines] [] (zip domain (map (normal 2 1) domain))
True
```



$$1 = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx$$

In [18]: values = [1, 1, 5]

In [19]: range values
Just (1,5)

In [20]: domain = [-4,-3.9..10]

```
In [21]: curve1 = map (normal 1 1) domain
```

```
In [22]: curve2 = map (normal 5 1) domain
```

```
In [23]: curves = [curve1, curve1, curve2]
```

```
In [24]: kde = foldl1 (zipWith (+)) curves
```

```
In [25]: plot X11 $ [Data2D [Style Lines] [] (zip domain kde)]
```

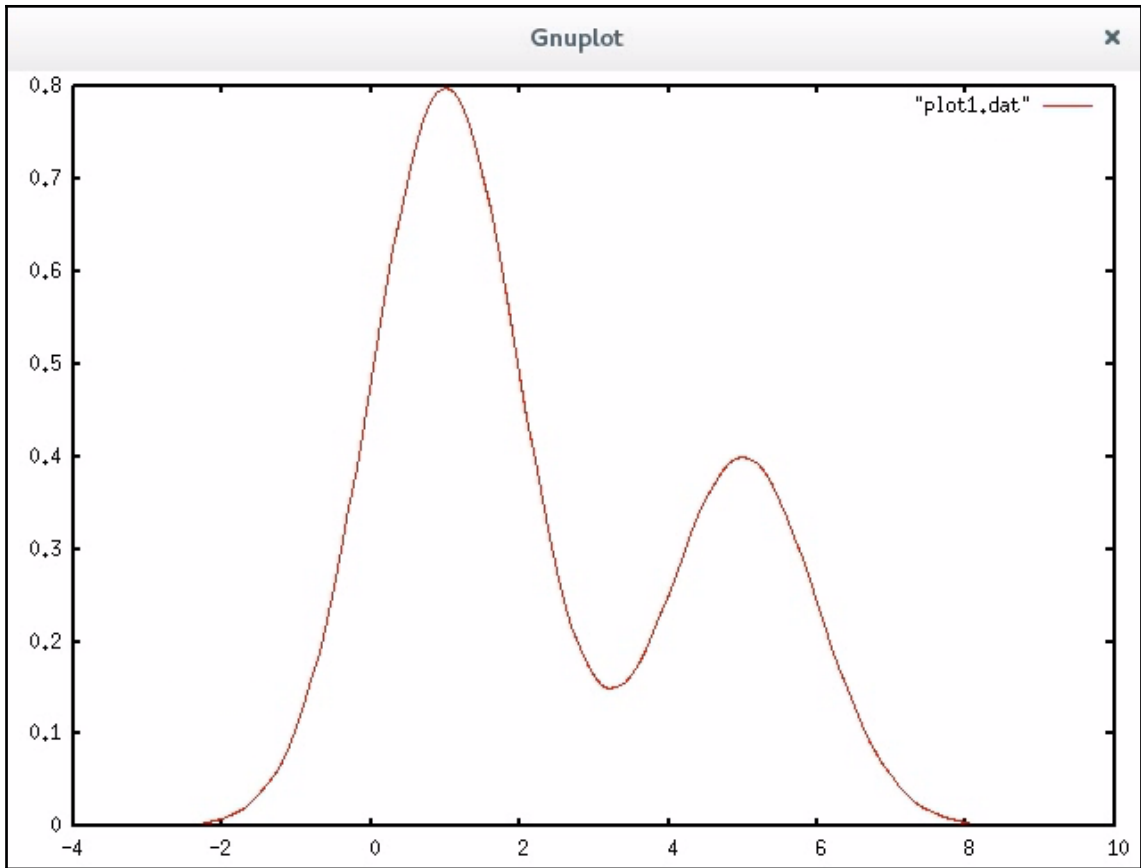
Redundant \$

Found:

```
plot X11 $ [Data2D [Style Lines] [] (zip domain kde)] plot X11 [Data2D [Style Lines] [] (zip domain kde)]
```

Why Not:

True



```
In [26]: kdeAdj = map (/ sum kde) kde
```

```
In [27]: sum kdeAdj
0.9999999999999994
```

```
In [28]: plot X11 $ [Data2D [Style Lines] [] (zip domain kdeAdj)]
```

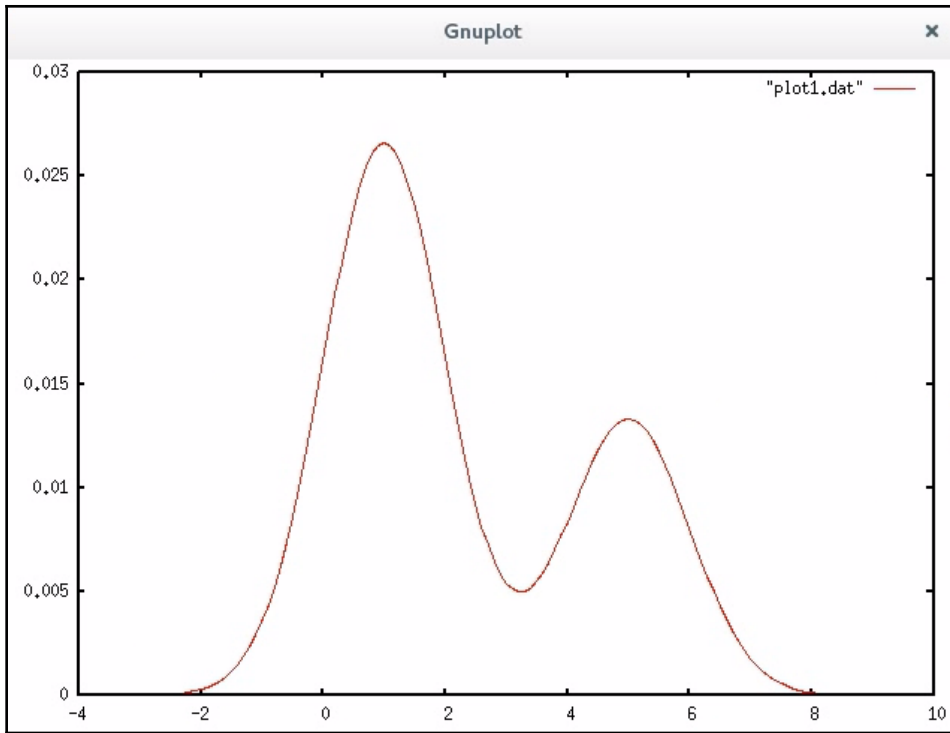
Redundant \$

Found:

Why Not:

```
plot X11 $ [Data2D [Style Lines] [] (zip domain kdeAdj)] plot X11 [Data2D [Style Lines] [] (zip domain kdeAdj)]
```







True



monet paintings

[All](#) [Images](#) [Shopping](#) [Videos](#) [Books](#) [More](#) [Search tools](#)

Claude Monet / Artwork

					
<p>Impression, Sunrise 1872</p>	<p>Woman with a Parasol - Mada... 1875</p>	<p>Poppies 1873</p>	<p>Women in the Garden 1866</p>	<p>Beach in Pourville 1882</p>	<p>The Water Lily Pond 1899</p>

monet paintings data set econometric

All Images Videos News More Settings Tools

About 1,24,000 results (0.58 seconds)

Econometric Analysis, 7th and 8th Edition, Data Sets - NYU Stern
pages.stern.nyu.edu/~wgreene/Text/Edition7/tablelist8new.htm ▼
 Table F4.2: Data on Sales of Monet Paintings (430 observations) Source: Table F5.2: Macroeconomics Data Set, Quarterly, 1950I to 2000IV, 204 Quarterly ...

[Table F4.1 \(csv format\)](#)

Table F4.2: Data on Sales of Monet Paintings (430 observations)
 Source: Author.

- Price = Sale Price in \$ (million),
- Height = Height (inches),
- Width = Width (inches),
- Signed = Dummy variable = 1 if signed, 0 if not,
- Picture = ID number (identifies repeat sales),
- House = Code for auction house where sale took place.

```

jcchurch@dataanalysis:~/Downloads$ ls
aapl.csv      earthquakes.csv  gl2015.zip      usgs.sqlite3
all_week.csv  GL2015.TXT      TableF4-1.csv
jcchurch@dataanalysis:~/Downloads$ mv TableF4-1.csv monet.csv
jcchurch@dataanalysis:~/Downloads$ vi monet.csv █
  
```

```
PRICE,HEIGHT,WIDTH,SIGNED,PICTURE,HOUSE ^M
3.9937800,21.3000000,25.6000000,1,1,1 ^M
8.8000000,31.9000000,25.6000000,1,2,2 ^M
.1316940,6.9000000,15.9000000,0,3,3 ^M
2.0375000,25.7000000,32,1,4,2 ^M
1.4875000,25.7000000,32,1,4,2 ^M
1.8700000,25.6000000,31.9000000,1,4,1 ^M
5.2825000,25.5000000,35.6000000,1,5,1 ^M
5.0657500,26,34.3000000,1,5,2 ^M
1.3750000,25.6000000,36.2000000,1,5,2 ^M
2.5300000,25.6000000,36.4000000,1,6,2 ^M
3.7425000,25.6000000,36.4000000,1,6,2 ^M
.3643430,25.6000000,36.2000000,1,7,2 ^M
2.7238700,31.9000000,39.4000000,1,8,2 ^M
3.5200000,23.6000000,31.9000000,1,9,1 ^M
.4975000,19.5000000,25,1,10,2 ^M
9.3500000,32.7000000,26.8000000,1,11,1 ^M
1.2195000,25.5000000,36,1,12,2 ^M
.4070000,25.6000000,39.4000000,1,12,2 ^M
3.7133900,25.6000000,36.2000000,1,13,3 ^M
5.2964060,25.8000000,21.5000000,1,14,2 ^M
2.8620000,21.4000000,28.9000000,1,15,2 ^M
"monet.csv" 431 lines, 16875 characters
```

```
jcchurch@dataanalysis:~/Downloads$ sed -i 's/^\./0./' monet.csv
```

```
3.9937800,21.3000000,25.6000000,1,1,1 ^M
8.8000000,31.9000000,25.6000000,1,2,2 ^M
0.1316940,6.9000000,15.9000000,0,3,3 ^M
2.0375000,25.7000000,32,1,4,2 ^M
1.4875000,25.7000000,32,1,4,2 ^M
1.8700000,25.6000000,31.9000000,1,4,1 ^M
```

```
jcchurch@dataanalysis:~/Downloads$ cp monet.csv ~/Code/HaskellDataAnalysis/data/
jcchurch@dataanalysis:~/Downloads$ █
```

```
In [29]: :load DescriptiveStats
         :load MyCSV
```

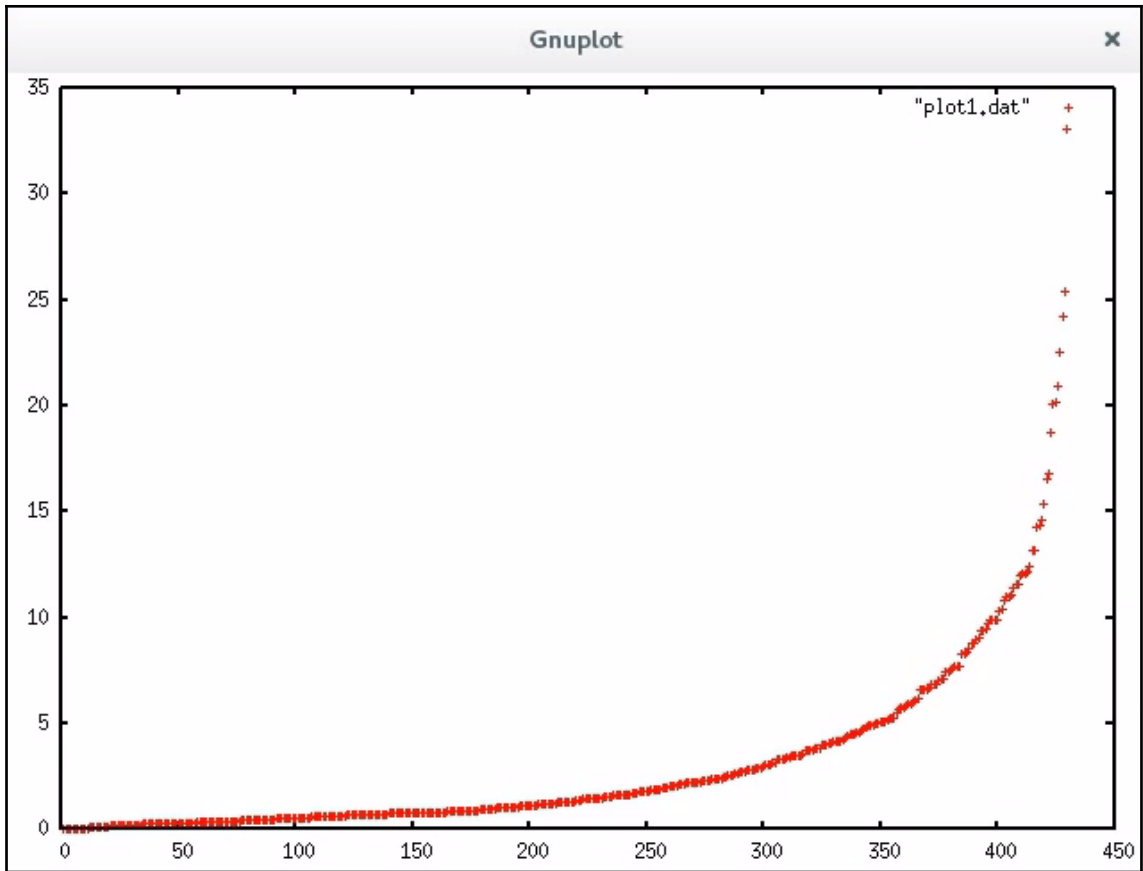
```
In [30]: import System.Random
         import Data.List
         import Data.Maybe
         import Text.CSV
         import Graphics.EasyPlot
         import DescriptiveStats
         import MyCSV
```

```
In [32]: kde :: [Double] -> Maybe ([Double], [Double])
         kde [] = Nothing
         kde xs = Just (domain, mykde)
         where
           mykde = map (/ sum shape) shape
           shape = foldl1 (zipWith (+)) (map (\x -> (map (normal x 1) domain)) xs)
           low = -5 + fst highlow
           high = 5 + snd highlow
           highlow = fromJust (range xs)
           domain = [low, (low+0.1)..high]
```

```
In [33]: monet <- parseCSVFromFile "../data/monet.csv"
```

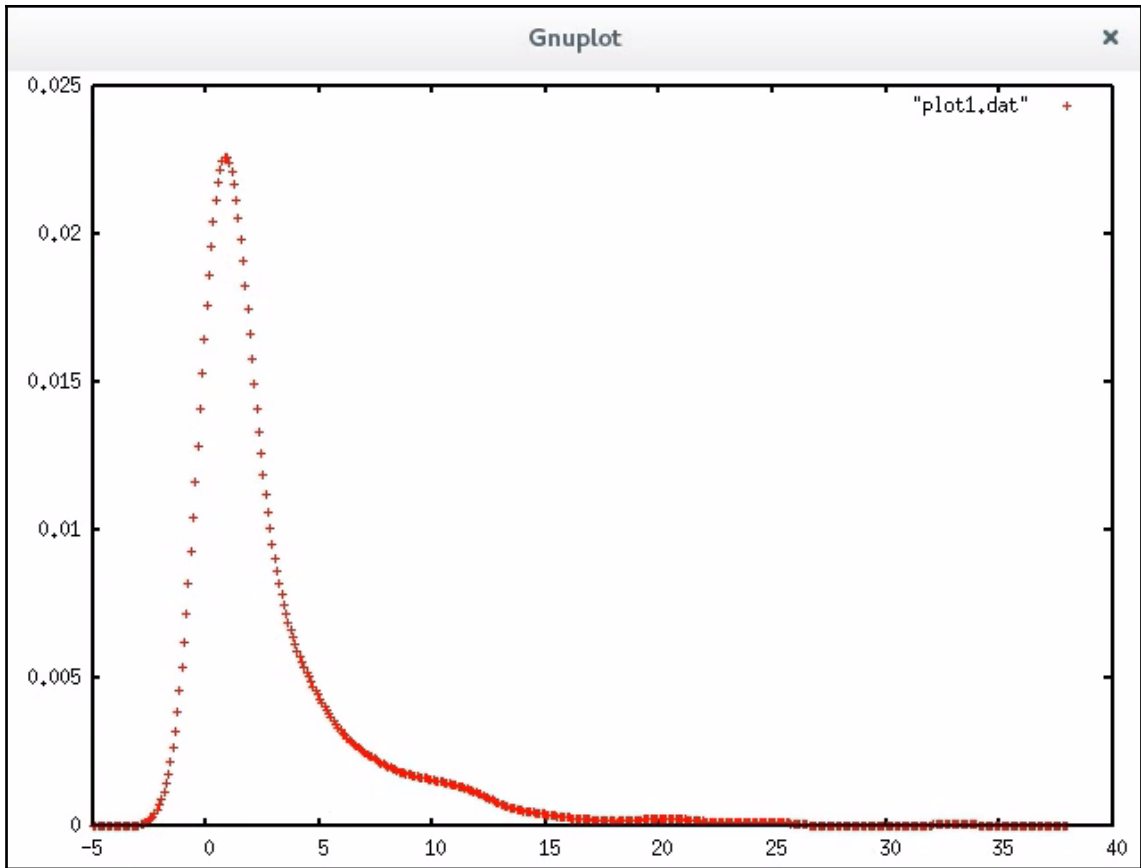
```
In [34]: monetPrices = readIndex monet 0 :: [Double]
```

```
In [35]: plot X11 $ zip [1..] (sort monetPrices)
         True
```



```
In [36]: Just (domain, kdeAdj) = kde monetPrices
```

```
In [37]: plot X11 $ zip domain kdeAdj
True
```



```
In [38]: indices = findIndices (>= 5) domain
```

```
In [39]: sum $ map (\x -> kdeAdj !! x) indices  
0.19975630172705675
```

Chapter 6: Course Review

MovieLens 100K Dataset

Stable benchmark dataset. 100,000 ratings from 1000 users on 1700 movies. Released 4/1998.

- [README.txt](#)
- [ml-100k.zip](#) (size: 5 MB, [checksum](#))
- [Index of unzipped files](#)

Permalink: <http://grouplens.org/datasets/movielens/100k/>

```
jcchurch@dataanalysis:~/Downloads$ ls
aapl.csv      earthquakes.csv  gl2015.zip    monet.csv
all_week.csv  GL2015.TXT     ml-100k.zip   usgs.sqlite3
jcchurch@dataanalysis:~/Downloads$ unzip ml-100k.zip █
```

```
jcchurch@dataanalysis:~/Downloads$ cd ml-100k/
jcchurch@dataanalysis:~/Downloads/ml-100k$ ls
allbut.pl  u1.base  u2.test  u4.base  u5.test  ub.base  u.genre  u.occupation
mku.sh     u1.test  u3.base  u4.test  ua.base  ub.test  u.info   u.user
README    u2.base  u3.test  u5.base  ua.test  u.data  u.item
jcchurch@dataanalysis:~/Downloads/ml-100k$ vi README █
```

DETAILED DESCRIPTIONS OF DATA FILES

=====

Here are brief descriptions of the data.

```
ml-data.tar.gz  -- Compressed tar file.  To rebuild the u data files do this:
gunzip ml-data.tar.gz
tar xvf ml-data.tar
mku.sh
```

106,0-1

65%

```
jchurch@dataanalysis: ~/Downloads/ml-100k
File Edit View Search Terminal Tabs Help
jchurch@dataanalysis: ~/Code/HaskellDataAnalysis
jchurch@dataanalysis: ~/Downloads/ml-100k

u.data -- The full u data set, 100000 ratings by 943 users on 1682 items.
        Each user has rated at least 20 movies. Users and items are
        numbered consecutively from 1. The data is randomly
        ordered. This is a tab separated list of
            user id | item id | rating | timestamp.
        The time stamps are unix seconds since 1/1/1970 UTC

u.info  -- The number of users, items, and ratings in the u data set.
        115,0-1 71%

jchurch@dataanalysis: ~/Downloads/ml-100k
File Edit View Search Terminal Help
196      242      3      881250949
186      302      3      891717742
22       377      1      878887116
244      51       2      880606923
166      346      1      886397596
298      474      4      884182806
115      265      2      881171488
"u.data" 100000L, 1979173C 1,1 Top
```

```
jchurch@dataanalysis:~/Downloads/ml-100k$ sqlite3 movies.sqlite3
SQLite version 3.8.7.1 2014-10-29 13:59:56
Enter ".help" for usage hints.
sqlite>
```

```
sqlite> CREATE TABLE data (userid INTEGER, itemid INTEGER, rating INTEGER, timestamp
INTEGER);
```

```
sqlite> .mode tabs
```

```
sqlite> .import u.data data
```

```
sqlite> SELECT * FROM data LIMIT 3;
196      242      3      881250949
186      302      3      891717742
22       377      1      878887116
```



```

u.occupation -- A list of the occupations.

u1.base      -- The data sets u1.base and u1.test through u5.base and u5.test
u1.test      are 80%/20% splits of the u data into training and test data.
u2.base      Each of u1, ..., u5 have disjoint test sets; this if for
u2.test      5 fold cross validation (where you repeat your experiment
u3.base      with each training and test set and average the results).
u3.test      These data sets can be generated from u.data by mku.sh.
u4.base
u4.test

```

144,1 91%

```

jcchurch@dataanalysis:~/Downloads/ml-100k$ ls
allbut.pl      u1.base  u3.base  u5.base  ub.base  u.info
mku.sh         u1.test  u3.test  u5.test  ub.test  u.item
movies.sqlite3 u2.base  u4.base  ua.base  u.data   u.occupation
README        u2.test  u4.test  ua.test  u.genre  u.user
jcchurch@dataanalysis:~/Downloads/ml-100k$ cp movies.sqlite3 ~/Code/HaskellDataAnalysis/data/
jcchurch@dataanalysis:~/Downloads/ml-100k$ ls ~/Code/HaskellDataAnalysis/data/
GL2015.TXT  monet.csv  movies.sqlite3  stocks.sqlite3  usgs.sqlite3
jcchurch@dataanalysis:~/Downloads/ml-100k$

```

```
In [1]: :load DescriptiveStats
```

```
In [2]: import Data.List
import Database.HDBC
import Database.HDBC.Sqlite3
import DescriptiveStats
```

```
In [3]: db <- connectSqlite3 "../data/movies.sqlite3"
```

```
In [4]: quickQuery db "SELECT avg(rating) FROM data" []
[[SqlDouble 3.52986]]
```

```
In [5]: quickQuery db "SELECT median(rating) FROM data" []
SqlError {seState = "", seNativeError = 1, seErrorMsg = "prepare 32: SELECT median(rating) FROM data: no such function: median"}
```

```
In [6]: quickQuery db "SELECT mode(rating) FROM data" []
SqlError {seState = "", seNativeError = 1, seErrorMsg = "prepare 30: SELECT mode(rating) FROM data: no such function: mode"}
```

```
In [7]: ratingsAction <- quickQuery db "SELECT data.rating FROM data, items WHERE data.itemid=items.movieid AND items.action=1" []
```

```
In [8]: length ratingsAction  
25589
```

```
In [9]: readColumn = map fromSql
```

```
In [10]: mean (readColumn ( (head . transpose) ratingsAction) :: [Double])  
Just 3.480245417953027
```

```
In [11]: median (readColumn ( (head . transpose) ratingsAction) :: [Double])  
Just 4.0
```

```
In [12]: ratingsDrama <- quickQuery db "SELECT data.rating FROM data, items WHERE data.itemid=items.movieid AND items.drama=1" []
```

```
In [13]: mean (readColumn ( (head . transpose) ratingsDrama) :: [Double])  
Just 3.6873793708484772
```

```
In [14]: median (readColumn ( (head . transpose) ratingsDrama) :: [Double])  
Just 4.0
```

```
In [15]: ratingsScifi <- quickQuery db "SELECT data.rating FROM data, items WHERE data.itemid=items.movieid AND items.scifi=1" []
```

```
In [16]: mean (readColumn ( (head . transpose) ratingsScifi) :: [Double])  
Just 3.5607227022780834
```

```
In [17]: median (readColumn ( (head . transpose) ratingsScifi) :: [Double])
Just 4.0
```

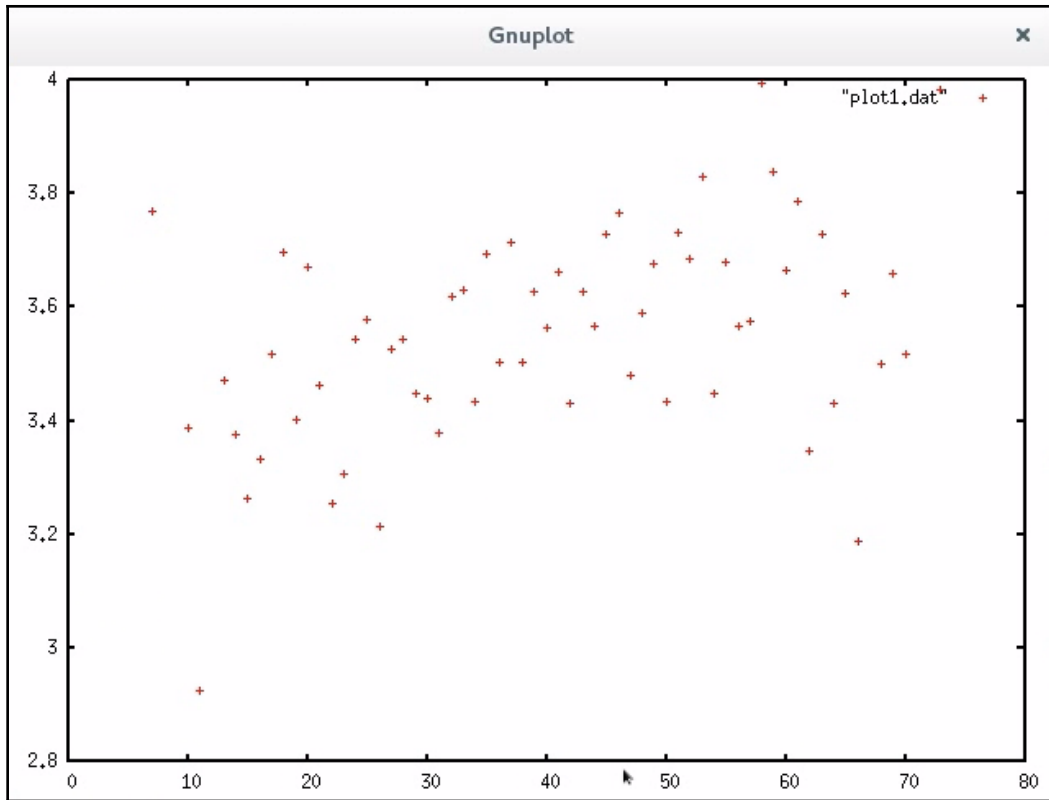
```
In [1]: :load DescriptiveStats

In [21]: import Data.List
import Data.Maybe
import Database.HDBC
import Database.HDBC.SQLite3
import Graphics.EasyPlot
import DescriptiveStats
```

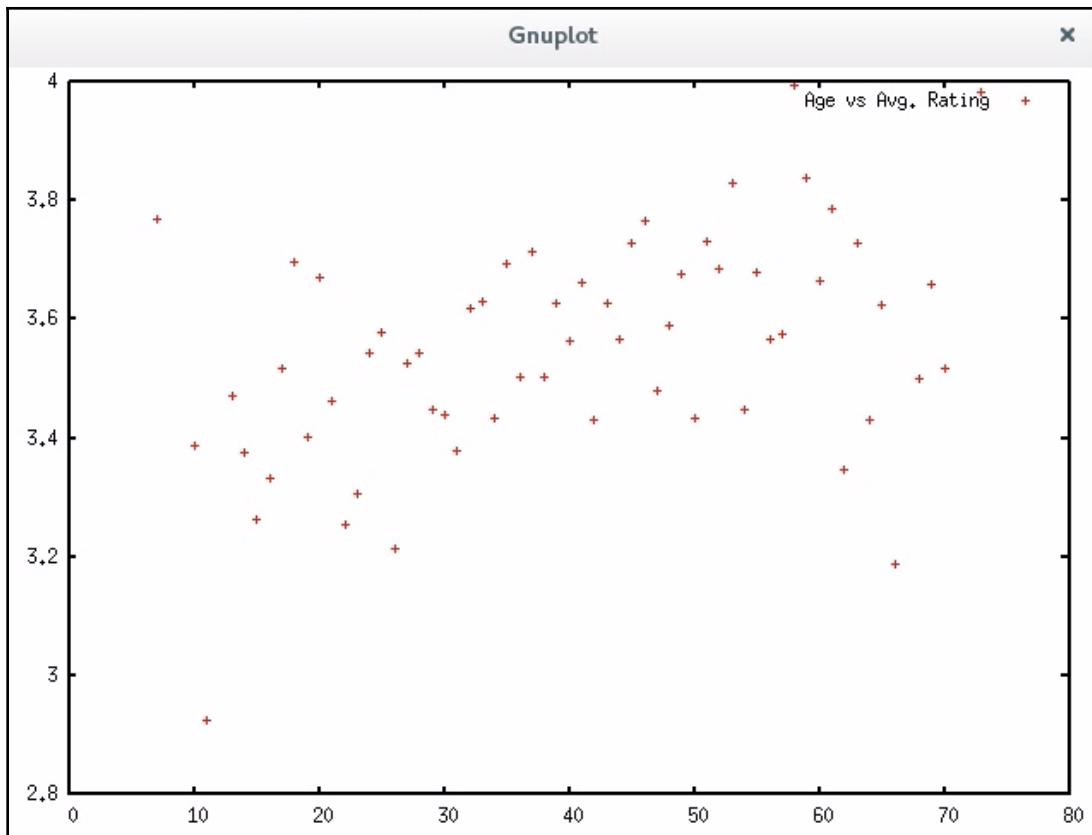
```
In [23]: age = readColumn (((!! 0) . transpose) ageRatings) :: [Double]
```

```
In [24]: avgRating = readColumn (((!! 1) . transpose) ageRatings) :: [Double]
```

```
In [25]: plot X11 $ zip age avgRating
True
```



```
In [22]: plot X11 $ Data2D [Title "Age vs Avg. Rating"] [] (zip age avgRating)
True
```



```
In [1]: ;load DescriptiveStats
```

```
In [33]: import Data.List
import Data.Maybe
import Database.HDBC
import Database.HDBC.SQLite3
import Graphics.EasyPlot
import DescriptiveStats
```

```
In [34]: kde :: [Double] -> Maybe ([Double], [Double])
kde [] = Nothing
kde xs = Just (domain, mykde)
  where
    mykde = map (/ sum shape) shape
    shape = foldl1 (zipWith (+)) (map (\x -> (map (normal x 1) domain)) xs)
    low = -5 + fst highlow
    high = 5 + snd highlow
    highlow = fromJust (range xs)
    domain = [low, (low+0.1)..high]
    normal mu sd x = exp (-((x-mu)^2 / (2*sd*sd))) / sqrt (2*sd*sd*pi)
```

```
[[SqlInt64 50,SqlByteString "Star Wars (1977)"],[SqlInt64 258,SqlByteString "Contact (1997)"],[SqlInt64 100,SqlByteString "
 Fargo (1996)"],[SqlInt64 181,SqlByteString "Return of the Jedi (1983)"],[SqlInt64 294,SqlByteString "Liar Liar (1997)"]
],[SqlInt64 286,SqlByteString "English Patient, The (1996)"],[SqlInt64 288,SqlByteString "Scream (1996)"],[SqlInt64 1,SqlB
yteString "Toy Story (1995)"],[SqlInt64 300,SqlByteString "Air Force One (1997)"],[SqlInt64 121,SqlByteString "Independen
ce Day (ID4) (1996)"]]
```

```
In [36]: starwarsRaw <- quickQuery db "SELECT rating FROM data WHERE itemid=50" []
```

```
In [37]: starwars = readColumn (((!! 0) . transpose) starwarsRaw) :: [Double]
```

```
In [38]: length starwars
583
```

```
In [39]: liarliarRaw <- quickQuery db "SELECT rating FROM data WHERE itemid=294" []
```

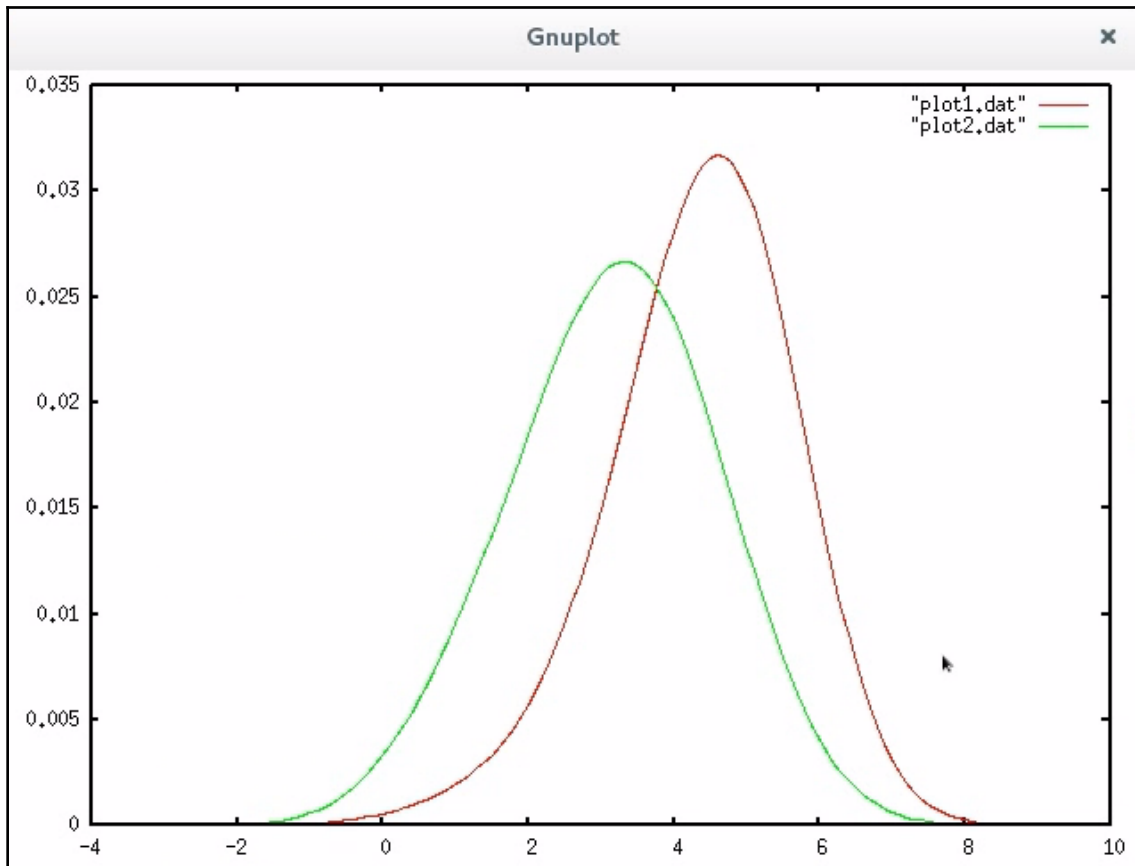
```
In [40]: liarliar = readColumn (((!! 0) . transpose) liarliarRaw) :: [Double]
```

```
In [41]: length liarliar
485
```

```
In [42]: Just (domainSw, mykdeSw) = kde starwars
```

```
In [43]: Just (domainLL, mykdeLL) = kde liarliar
```

```
In [45]: plot X11 [Data2D [Style Lines] []] (zip domainSW mykdeSW), Data2D [Style Lines] [] (zip domainLL mykdeLL)]  
True
```



In [46]: domainSw

```
[-4.0, -3.9, -3.8, -3.699999999999997, -3.599999999999996, -3.499999999999996, -3.399999999999995, -3.299999999999994, -3.19
999999999999993, -3.099999999999999, -2.999999999999999, -2.899999999999999, -2.799999999999999, -2.699999999999999, -2.59999999
9999998, -2.499999999999998, -2.399999999999998, -2.299999999999998, -2.199999999999998, -2.099999999999998, -1.999999999
9999982, -1.899999999999981, -1.79999999999998, -1.69999999999998, -1.599999999999979, -1.499999999999978, -1.3999999999999
977, -1.299999999999976, -1.199999999999975, -1.099999999999974, -0.999999999999973, -0.899999999999972, -0.7999999999999
972, -0.699999999999971, -0.59999999999997, -0.499999999999969, -0.399999999999968, -0.299999999999967, -0.199999999999966
62, -9.99999999999654e-2, 3.552713678800501e-15, 0.10000000000000364, 0.20000000000000373, 0.3000000000000038, 0.40000000000000
039, 0.500000000000004, 0.6000000000000041, 0.7000000000000042, 0.8000000000000043, 0.9000000000000044, 1.0000000000000044, 1.10
00000000000045, 1.2000000000000046, 1.3000000000000047, 1.4000000000000048, 1.5000000000000049, 1.600000000000005, 1.7000000000
0005, 1.800000000000052, 1.900000000000052, 2.000000000000053, 2.100000000000054, 2.200000000000055, 2.300000000000056, 2
.400000000000057, 2.500000000000058, 2.60000000000006, 2.70000000000006, 2.80000000000006, 2.90000000000006, 3.0000000000
0006, 3.100000000000063, 3.200000000000064, 3.300000000000065, 3.400000000000066, 3.500000000000067, 3.600000000000068, 3
.70000000000007, 3.80000000000007, 3.90000000000007, 4.00000000000007, 4.10000000000007, 4.20000000000006, 4.30000000000006, 4.40000000000006, 4.50000000000005, 4.60000000000005, 4.70000000000005, 4.80000000000004, 4.90000000000004, 5.00000000
000036, 5.10000000000003, 5.20000000000003, 5.300000000000025, 5.40000000000002, 5.50000000000002, 5.60000000000001, 5.70000000000001, 5.80000000000001, 5.9, 6.0, 6.1, 6.19999999999999, 6.29999999999999, 6.39999999999999, 6.49999999999998, 6.5
9999999999998, 6.69999999999975, 6.7999999999997, 6.8999999999997, 6.99999999999964, 7.0999999999996, 7.199999999999
96, 7.2999999999995, 7.3999999999995, 7.4999999999995, 7.5999999999994, 7.6999999999994, 7.7999999999994, 7.89999999
999993, 7.9999999999993, 8.0999999999993, 8.1999999999992, 8.2999999999992, 8.3999999999991, 8.4999999999991, 8.59
99999999999, 8.6999999999999, 8.7999999999999, 8.8999999999999, 8.9999999999999, 9.0999999999999, 9.1999999999999, 9.29
9999999999988, 9.39999999999988, 9.4999999999998, 9.59999999999987, 9.69999999999987, 9.79999999999986, 9.89999999999986
, 9.99999999999986]
```

In [47]: domainLL

```
[-4.0, -3.9, -3.8, -3.699999999999997, -3.599999999999996, -3.499999999999996, -3.399999999999995, -3.299999999999994, -3.19
999999999999993, -3.099999999999999, -2.999999999999999, -2.899999999999999, -2.799999999999999, -2.699999999999999, -2.59999999
9999998, -2.499999999999998, -2.399999999999998, -2.299999999999998, -2.199999999999998, -2.099999999999998, -1.999999999
9999982, -1.899999999999981, -1.79999999999998, -1.69999999999998, -1.599999999999979, -1.499999999999978, -1.3999999999999
977, -1.299999999999976, -1.199999999999975, -1.099999999999974, -0.999999999999973, -0.899999999999972, -0.7999999999999
972, -0.699999999999971, -0.59999999999997, -0.499999999999969, -0.399999999999968, -0.299999999999967, -0.199999999999966
62, -9.99999999999654e-2, 3.552713678800501e-15, 0.10000000000000364, 0.20000000000000373, 0.3000000000000038, 0.40000000000000
039, 0.500000000000004, 0.6000000000000041, 0.7000000000000042, 0.8000000000000043, 0.9000000000000044, 1.0000000000000044, 1.10
00000000000045, 1.2000000000000046, 1.3000000000000047, 1.4000000000000048, 1.5000000000000049, 1.600000000000005, 1.7000000000
0005, 1.800000000000052, 1.900000000000052, 2.000000000000053, 2.100000000000054, 2.200000000000055, 2.300000000000056, 2
.400000000000057, 2.500000000000058, 2.60000000000006, 2.70000000000006, 2.80000000000006, 2.90000000000006, 3.00000000000006, 3.100000000000063, 3.200000000000064, 3.300000000000065, 3.400000000000066, 3.500000000000067, 3.600000000000068, 3
.70000000000007, 3.80000000000007, 3.90000000000007, 4.00000000000007, 4.10000000000007, 4.20000000000006, 4.30000000000006, 4.40000000000006, 4.50000000000005, 4.60000000000005, 4.70000000000005, 4.80000000000004, 4.90000000000004, 5.00000000
000036, 5.10000000000003, 5.20000000000003, 5.300000000000025, 5.40000000000002, 5.50000000000002, 5.60000000000001, 5.70000000000001, 5.80000000000001, 5.9, 6.0, 6.1, 6.19999999999999, 6.29999999999999, 6.39999999999999, 6.49999999999998, 6.5
9999999999998, 6.69999999999975, 6.7999999999997, 6.8999999999997, 6.99999999999964, 7.0999999999996, 7.199999999999
96, 7.2999999999995, 7.3999999999995, 7.4999999999995, 7.5999999999994, 7.6999999999994, 7.7999999999994, 7.89999999
999993, 7.9999999999993, 8.0999999999993, 8.1999999999992, 8.2999999999992, 8.3999999999991, 8.4999999999991, 8.59
99999999999, 8.6999999999999, 8.7999999999999, 8.8999999999999, 8.9999999999999, 9.0999999999999, 9.1999999999999, 9.29
9999999999988, 9.39999999999988, 9.4999999999998, 9.59999999999987, 9.69999999999987, 9.79999999999986, 9.89999999999986
, 9.99999999999986]
```

In [48]: `sum $ map (\(sw, ll) -> if ll <= sw then ll else sw) (zip mykdeSw mykdeLL)`
0.6533607763536907

In [49]: `sum $ map (\(sw, ll) -> if ll <= sw then ll else sw) (zip mykdeSw mykdeSw)`
1.000000000000002