CS 696 Intro to Big Data: Tools and Methods
Fall Semester, 2016
Doc 10 Statistics
Sep 26, 2016

Descriptive Statistics

mean
median
mode
variance
standard variation
quantiles

Descriptive Statistics

Arithmetic mean

mean(numbers) = sum(numbers)/length(numbers)

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

mean([1,7,3,8,5]) == 4.80

median

- Middle value of sorted list of numbers
- If even number of values then mean of middle two values

median([1,7,3,8,5]) == 5.00

mode

- Value that appears the most in the data

Descriptive Statistics

Variance

Measures the spread in the numbers

$$s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2$$

Standard Deviation, (SD, s, σ)

square root of the variance

Bessel's Correction

Normally only have a sample of data

$$s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2$$

Computing mean from sample introduces bias

Bessel's correction for this bias

Divide by N-1

$$s^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2.$$

For large N this is not needed

But if underlying distribution is skewed or has long tails (kurtosis) other biases are introduced

Julia functions

Use Bessel's correction

var([2,4,4,4,5,5,7,9])	4.57
std([2,4,4,4,5,5,7,9])	2.14

var([2,4,4,4,5,5,7,9],mean=5)	4.57
std([2,4,4,4,5,5,7,9],mean=9)	4.78

Me & Bill Gates

mean of mine & Bill Gates net worth = $39.6 B

variance 3144.2
standard deviation 51.6

mean of Zuckerberg & Carlos Slim net worth = $52.3 B

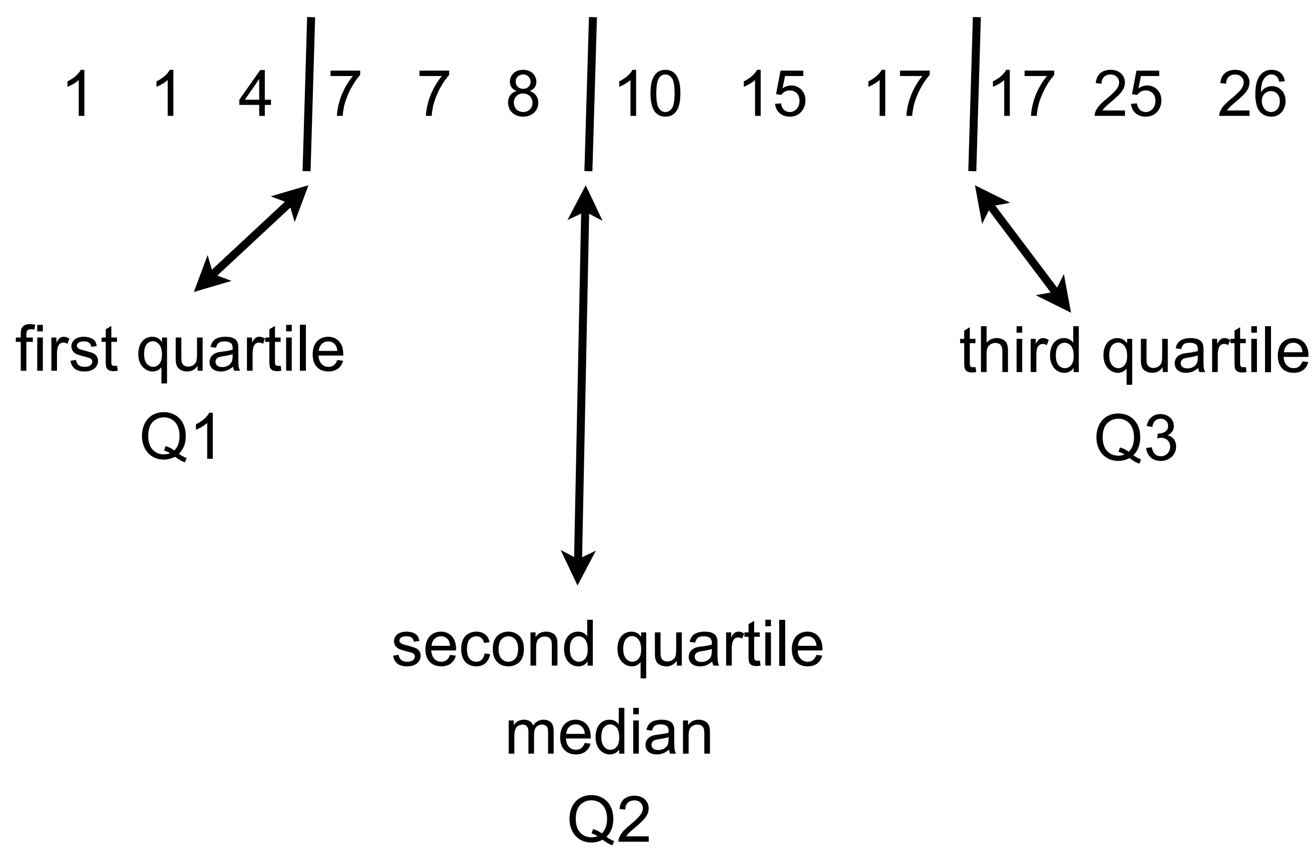
variance 11.5
standard deviation 3.39

Quantiles

q-quantiles

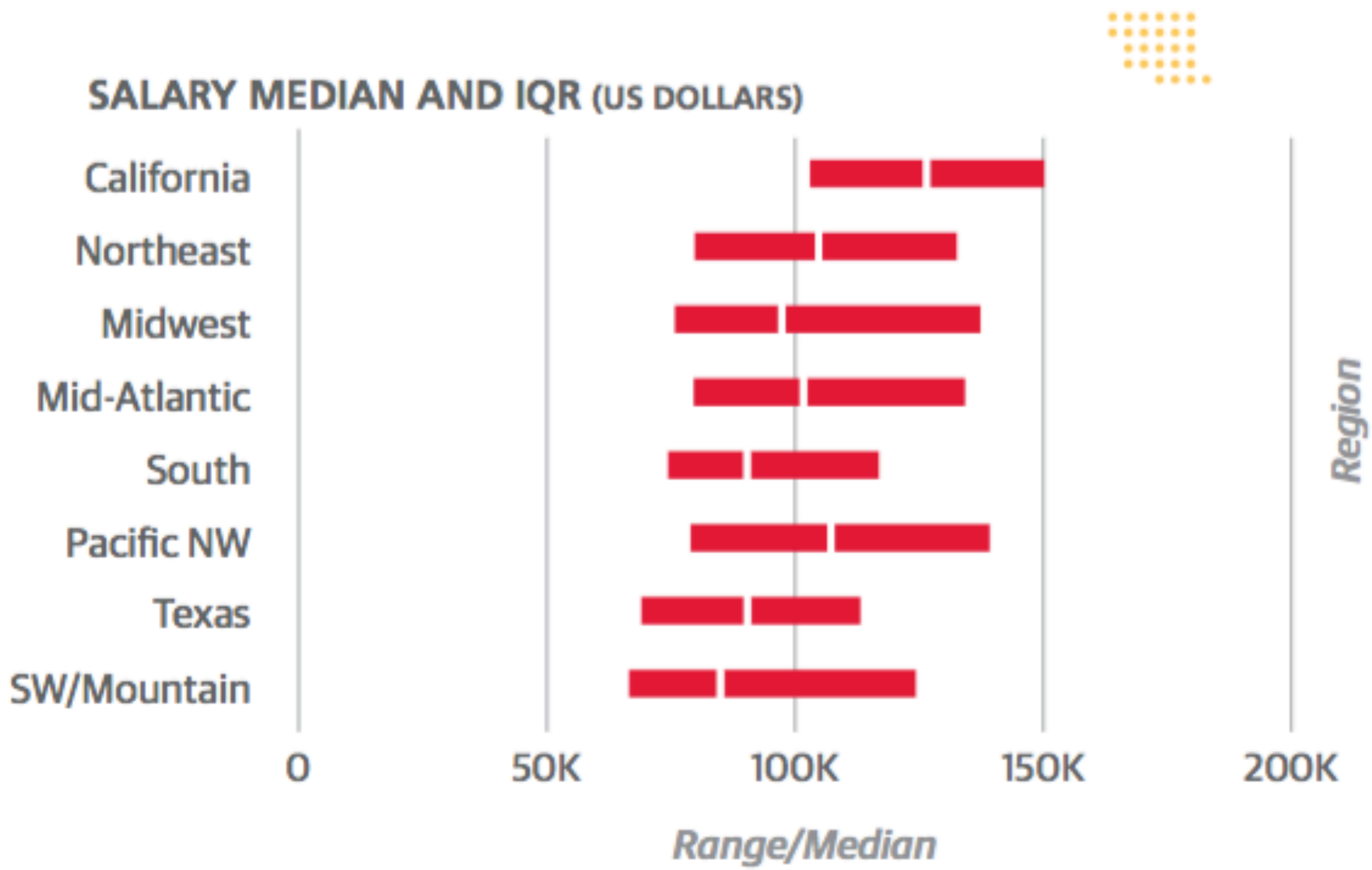
Cutpoints that divide the sorted data into q equal sized groups

4-quantile, quartile



Red Bar shows middle two quartiles

White bar is median

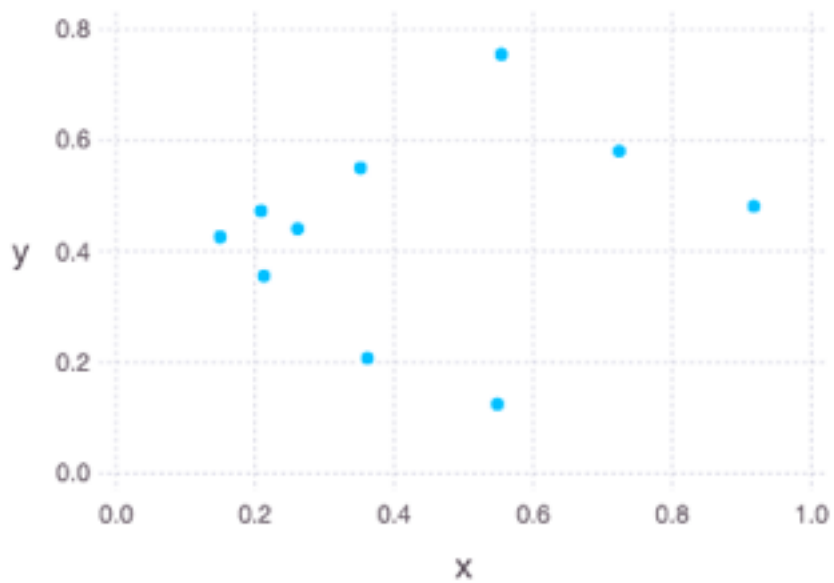


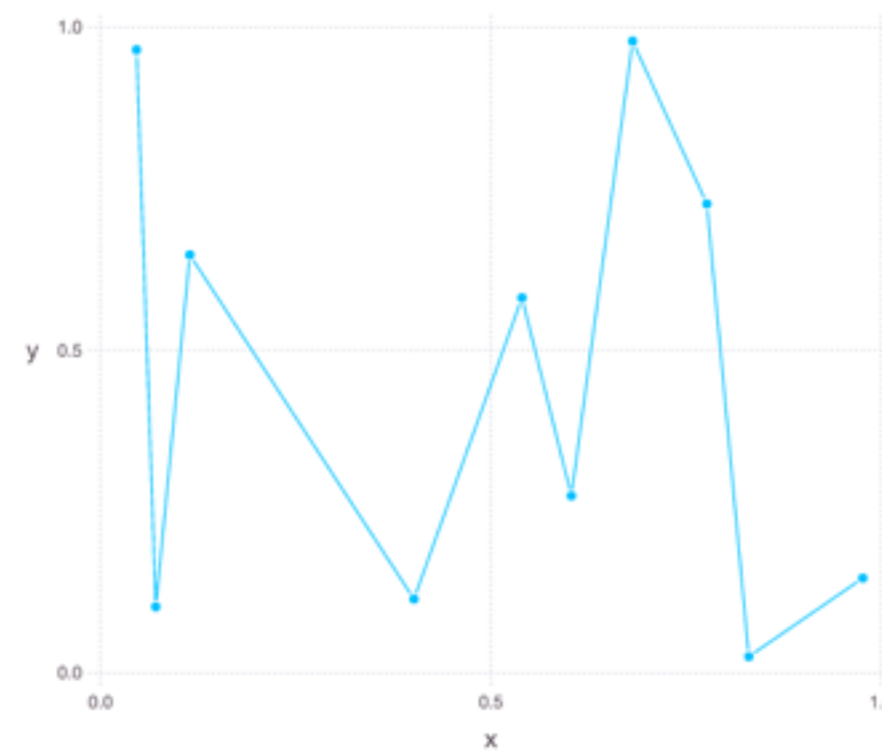
Plotting with Gadfly

http://gadflyjl.org/stable/index.html

using Gadfly

plot(x=rand(10), y=rand(10))





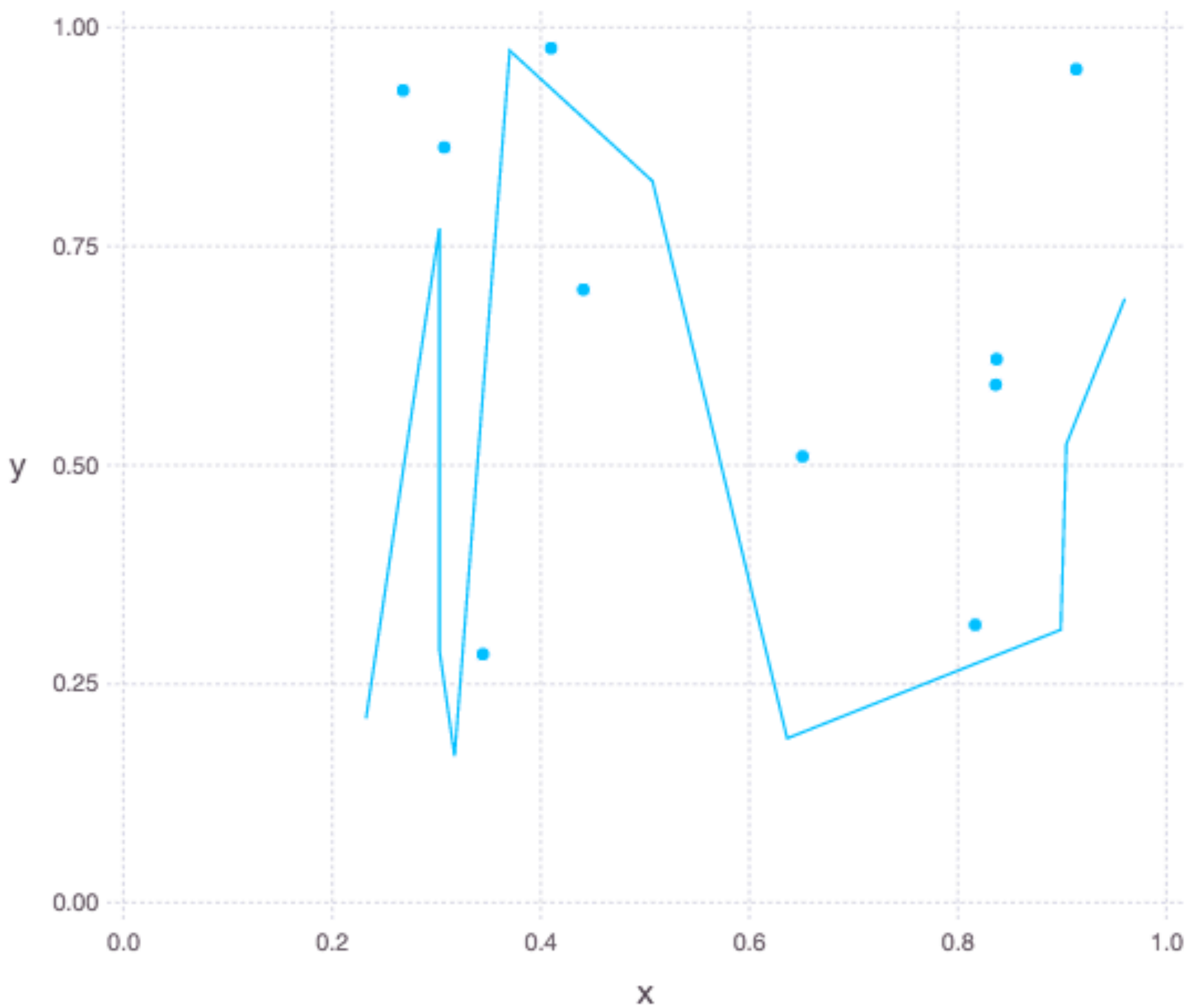
plot(x=rand(10), y=rand(10), Geom.point, Geom.line)

Gadfly Features

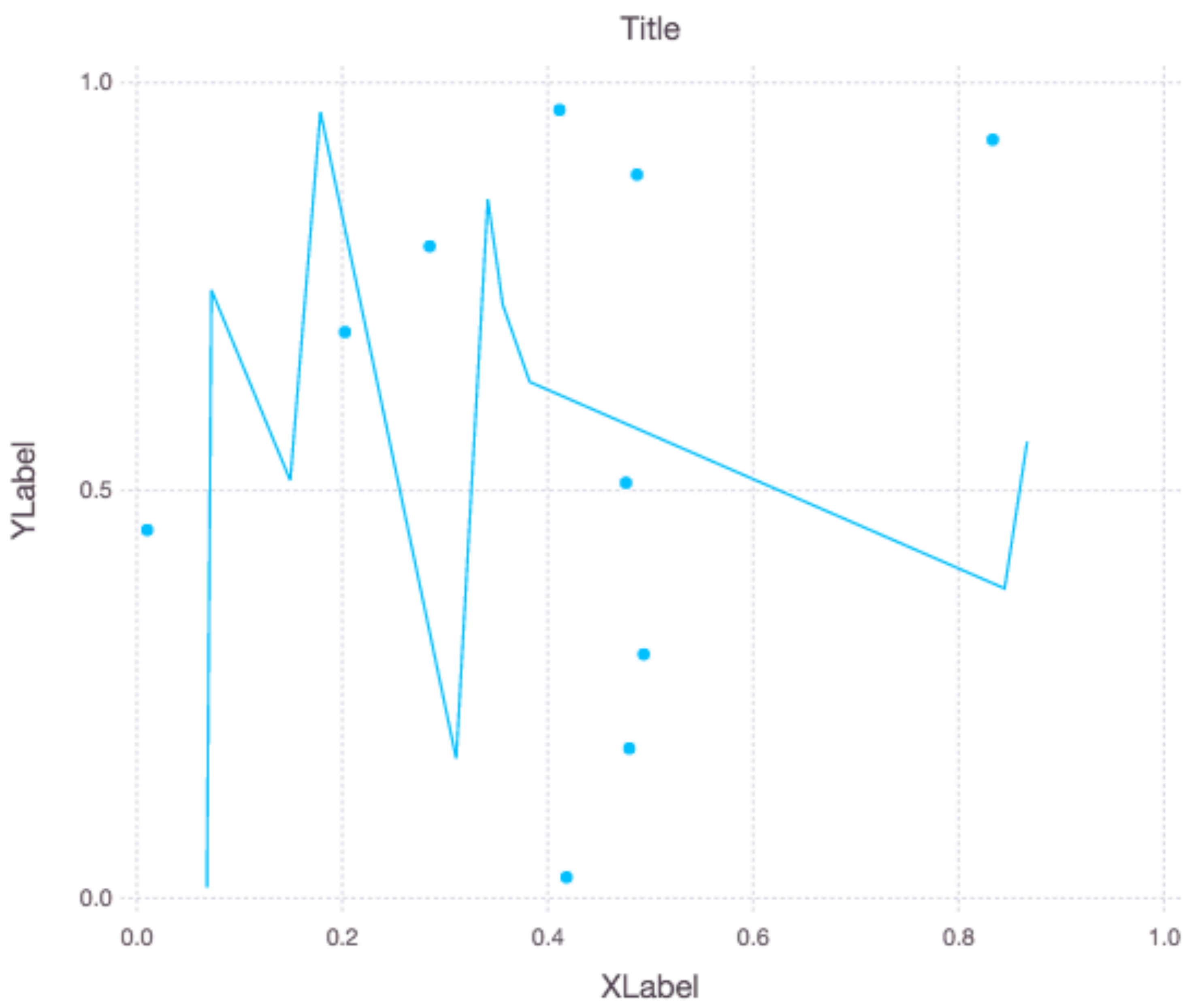
Layers
Themes
Geometries
Guides
Statistics
Scales

Layers

```
plot(layer(x=rand(10), y=rand(10), Geom.point),
     layer(x=rand(10), y=rand(10), Geom.line))
```

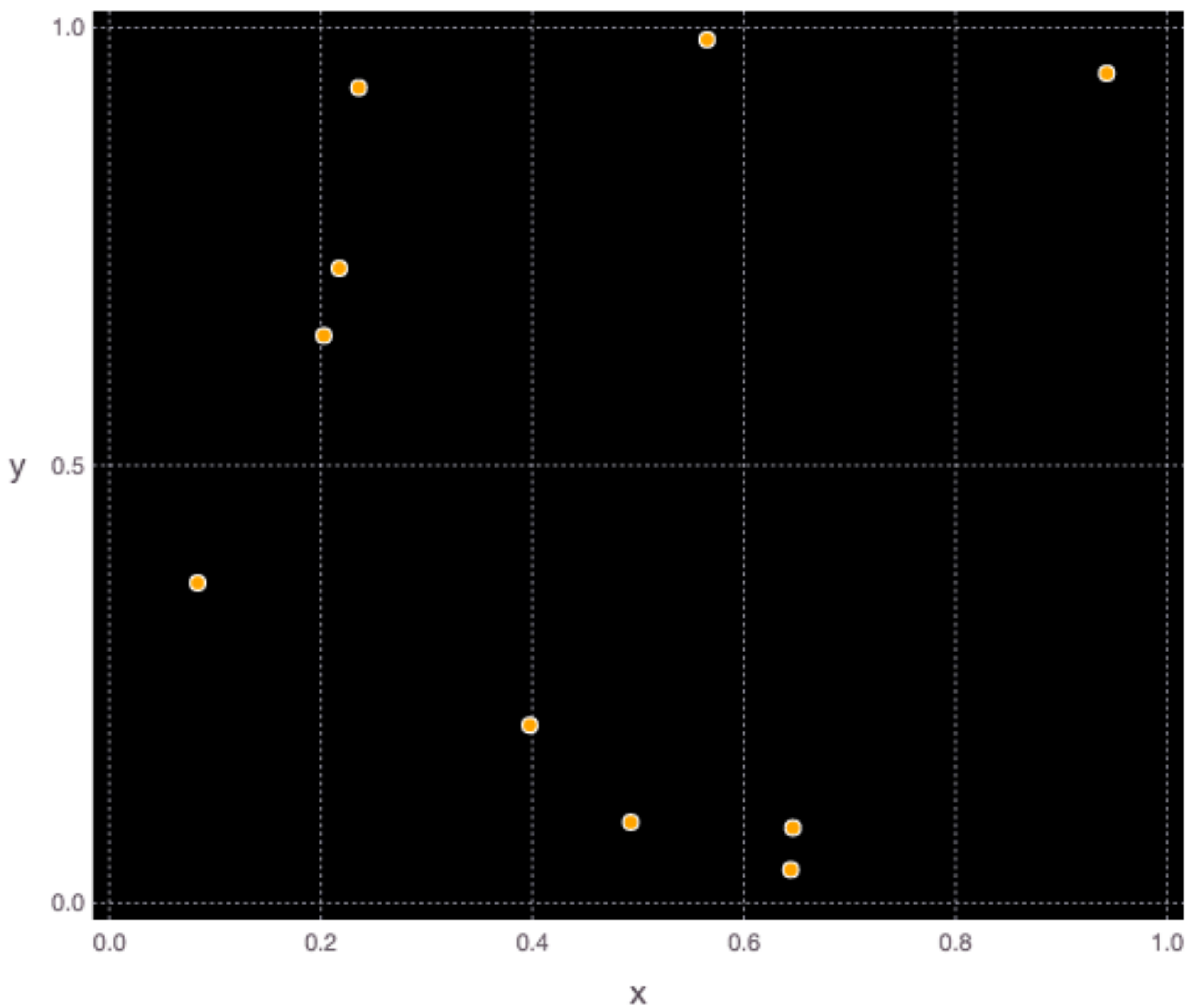


```
plot(layer(x=rand(10), y=rand(10), Geom.point, order = 2),
    layer(x=rand(10), y=rand(10), Geom.line, order = 1),
    Guide.XLabel("XLabel"),
    Guide.YLabel("YLabel"),
    Guide.Title("Title"))
```



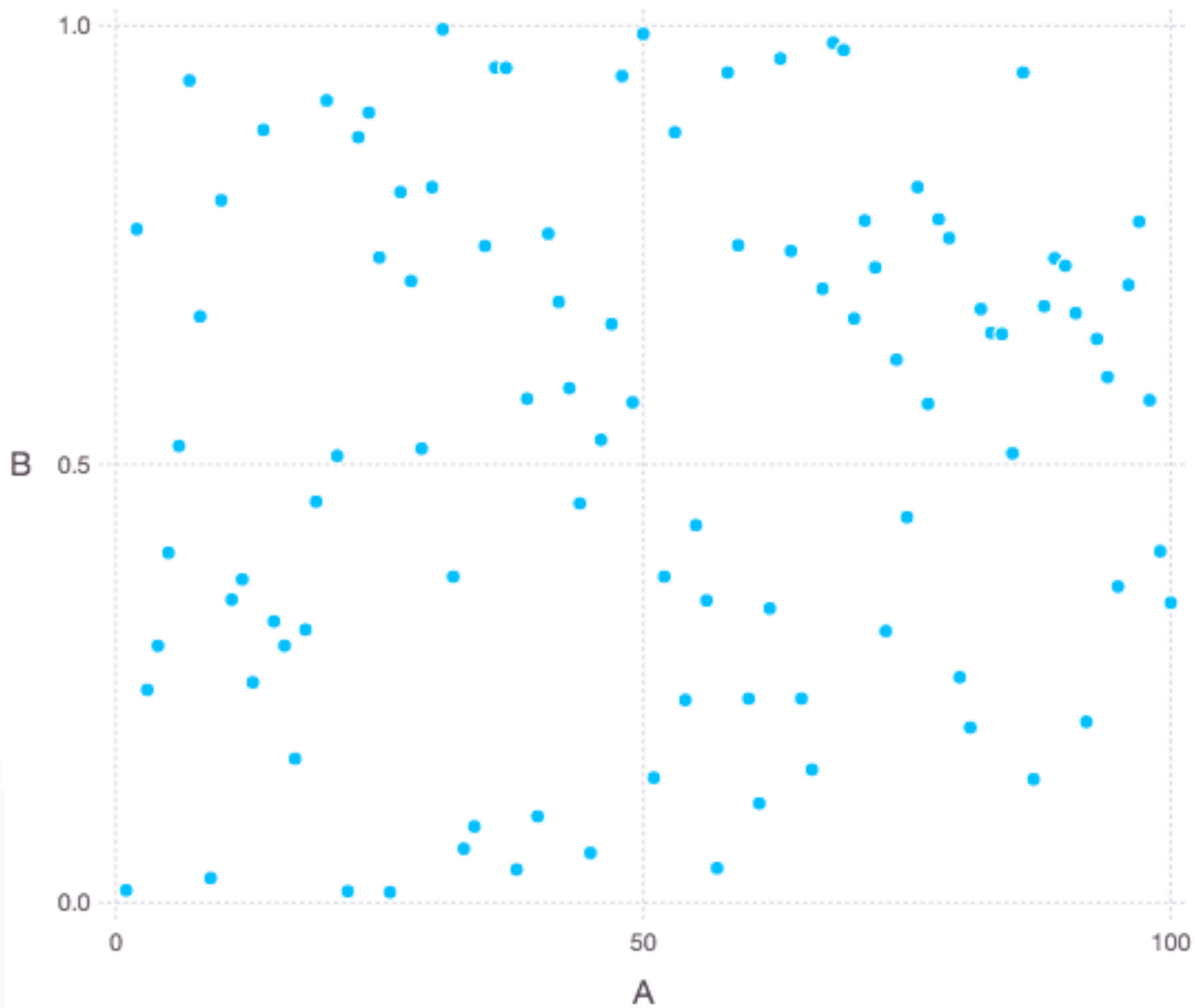
Themes

```
plot(x=rand(10), y=rand(10),
   Theme(panel_fill=colorant"black", default_color=colorant"orange"))
```



Using DataFrames

```
large = DataFrame(A = 1:100, B = rand(100))
plot(large, x = "A", y = "B")
```



R Datasets

Datasets collected to use to learn statistics & use R

Commonly used

List

https://vincentarelbundock.github.io/Rdatasets/datasets.html

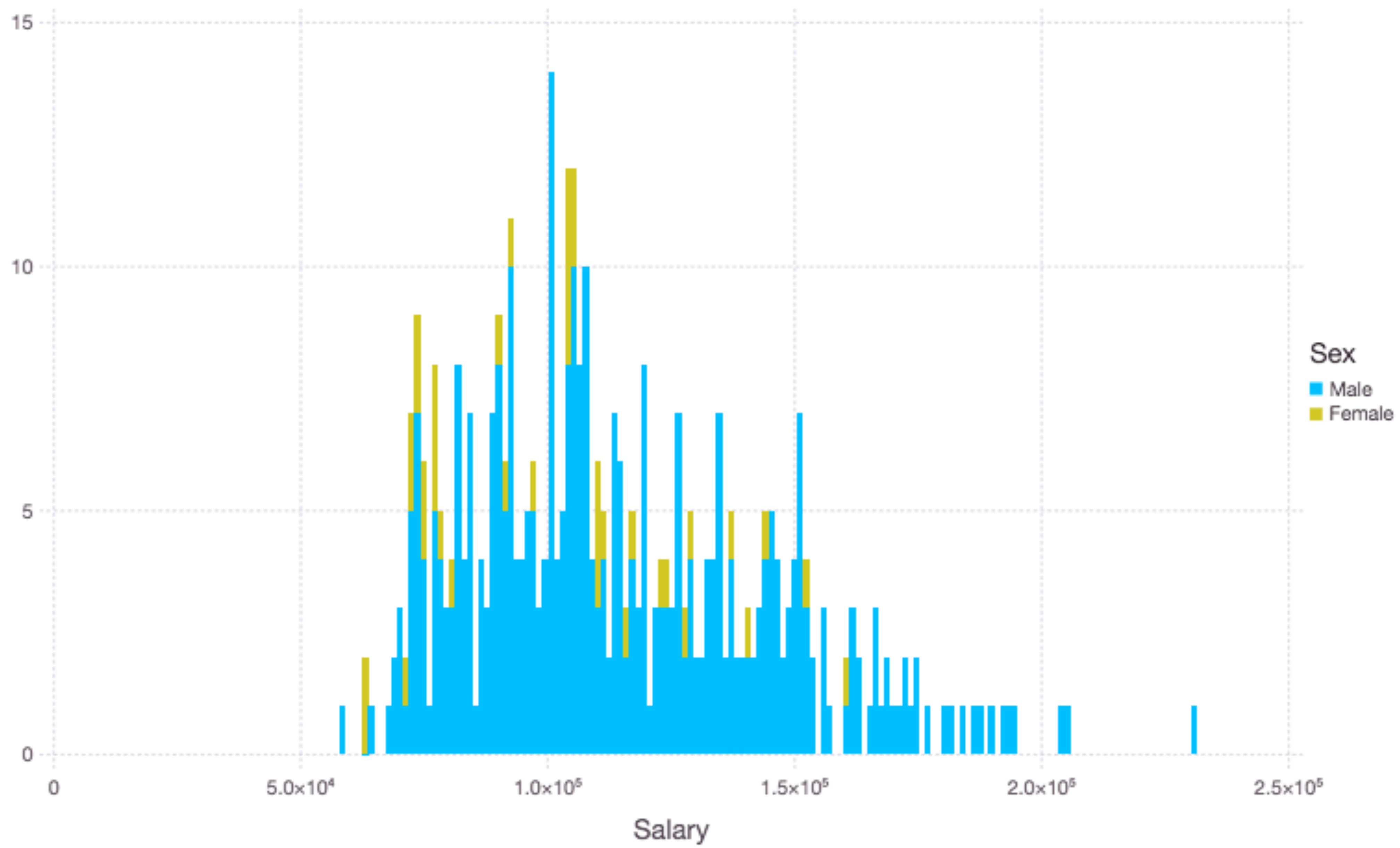
```
using DataFrames
using RDatasets

dataset("car", "Salaries")
```

2008-9 Academic Salary

```
397×6 DataFrames.DataFrame
| Row | Rank| Discipline | YrsSincePhD | YrsService | Sex |   Salary |
| 1   | "Prof"| "B"      | 19          | 18         | "Male" | 139750 |
| 2   | "Prof"| "B"      | 20          | 16         | "Male" | 173200 |
```

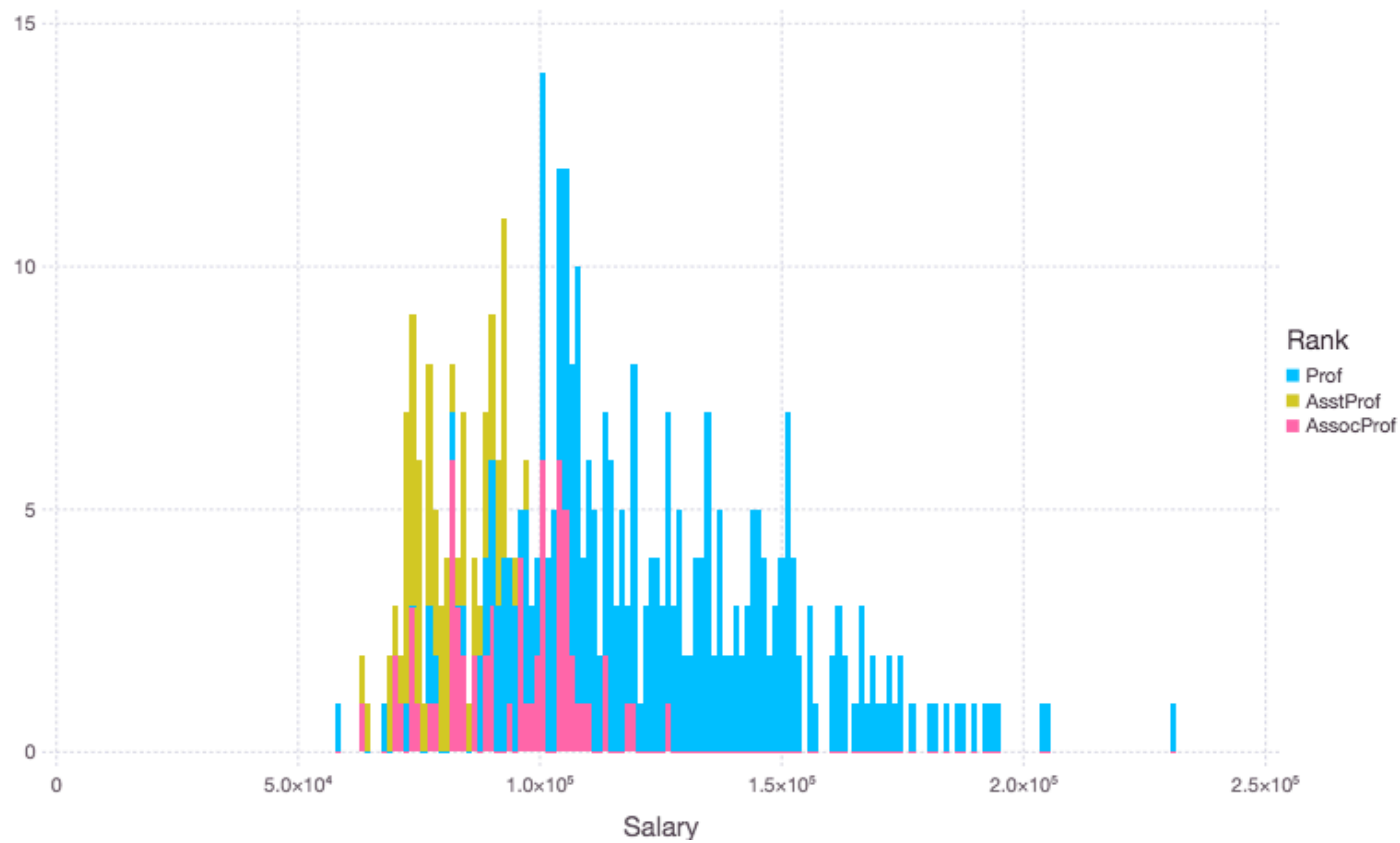
Salary & Sex

```
plot(dataset("car", "Salaries"), x="Salary", color="Sex", Geom.histogram)
```



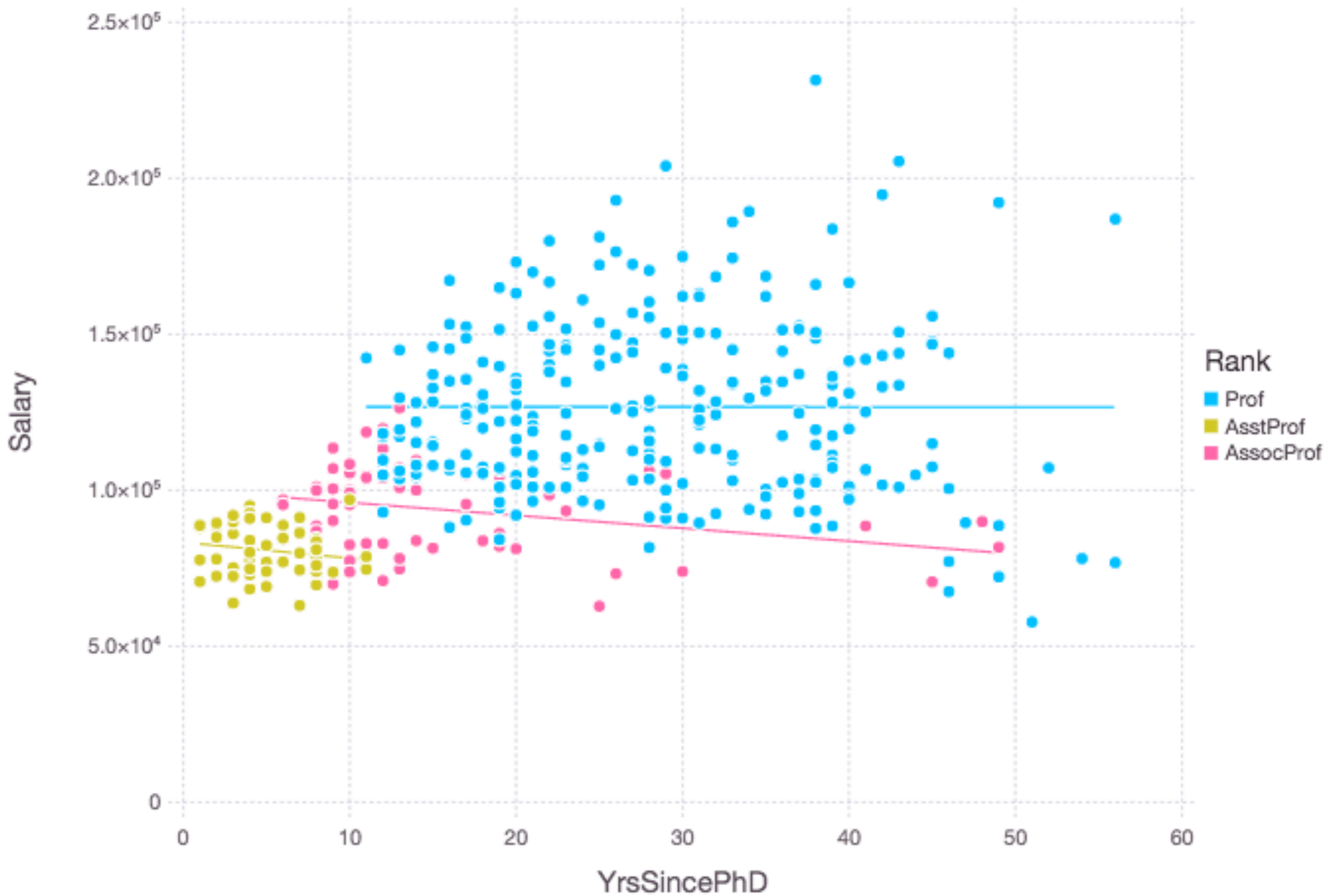
Salary & Rank

plot(dataset("car", "Salaries"), x="Salary", color="Rank", Geom.histogram)



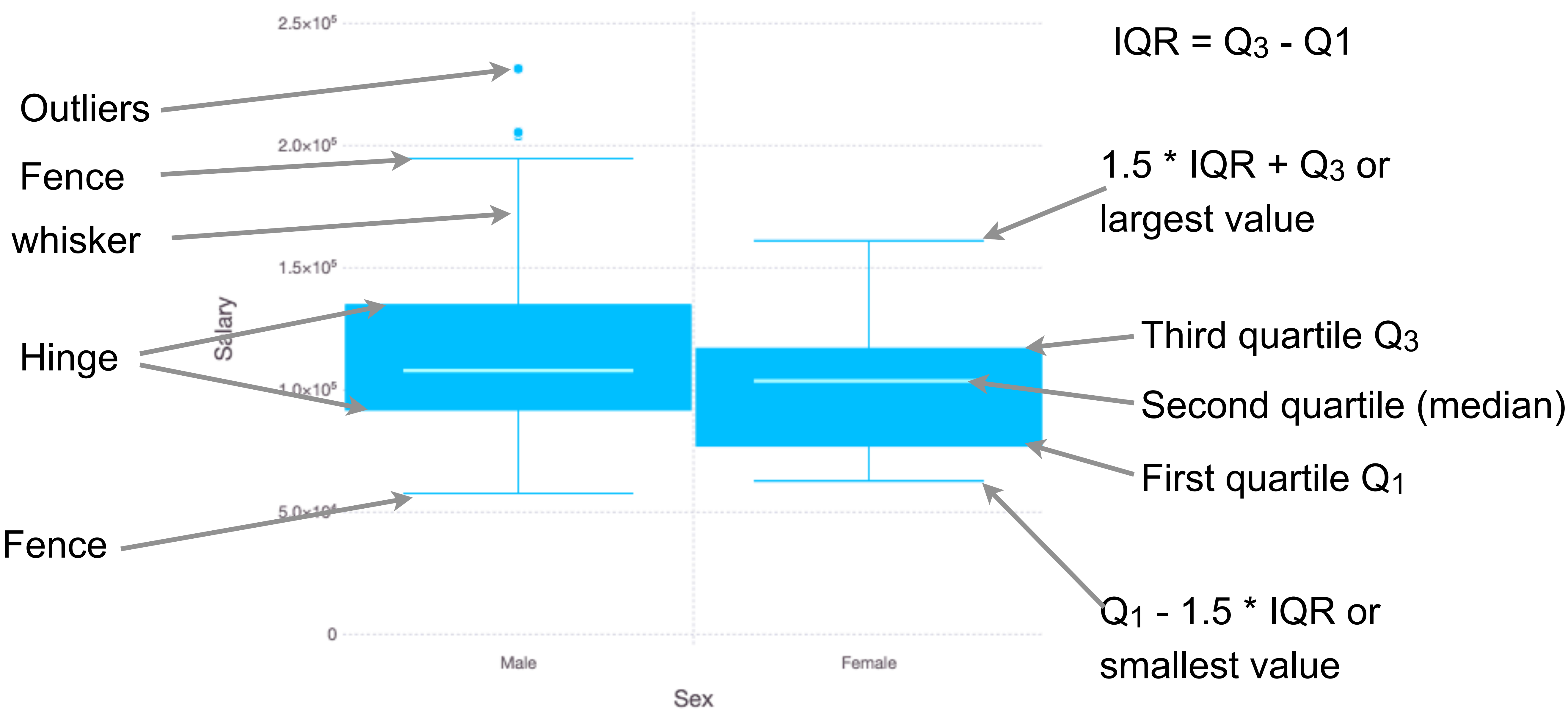
Scatter Plot: Salary-Years Colored by Rank

```
plot(dataset("car", "Salaries"), y="Salary", x="YrsSincePhD", color="Rank",
        Geom.point,
        Geom.smooth(method=:lm))
```



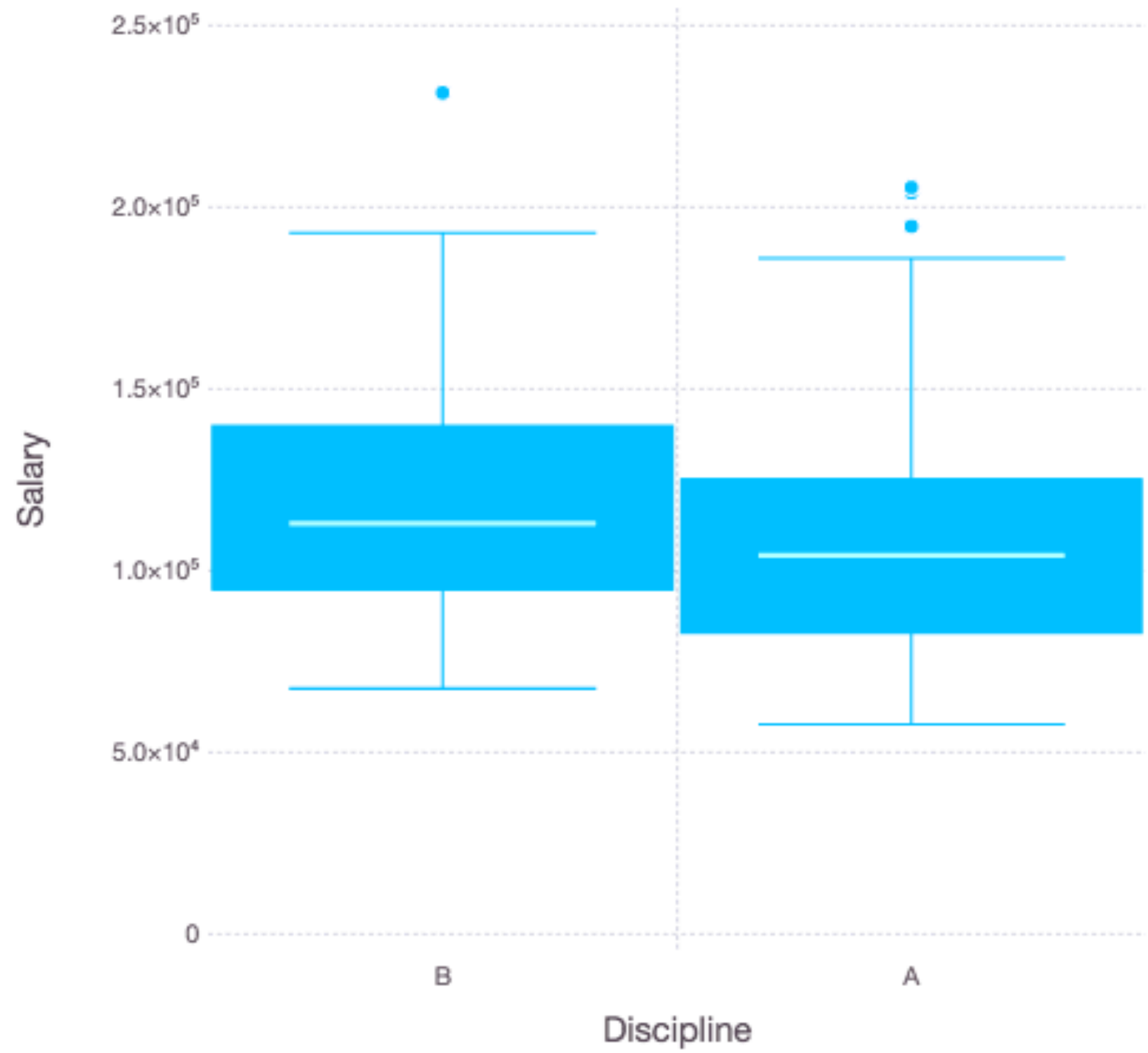
Box Plots (Tukey Method)

```
plot(dataset("car", "Salaries"), y="Salary", x="Sex", Geom.boxplot)
```



Salary by Discipline

plot(dataset("car", "Salaries"), y="Salary", x="Discipline",Geom.boxplot)

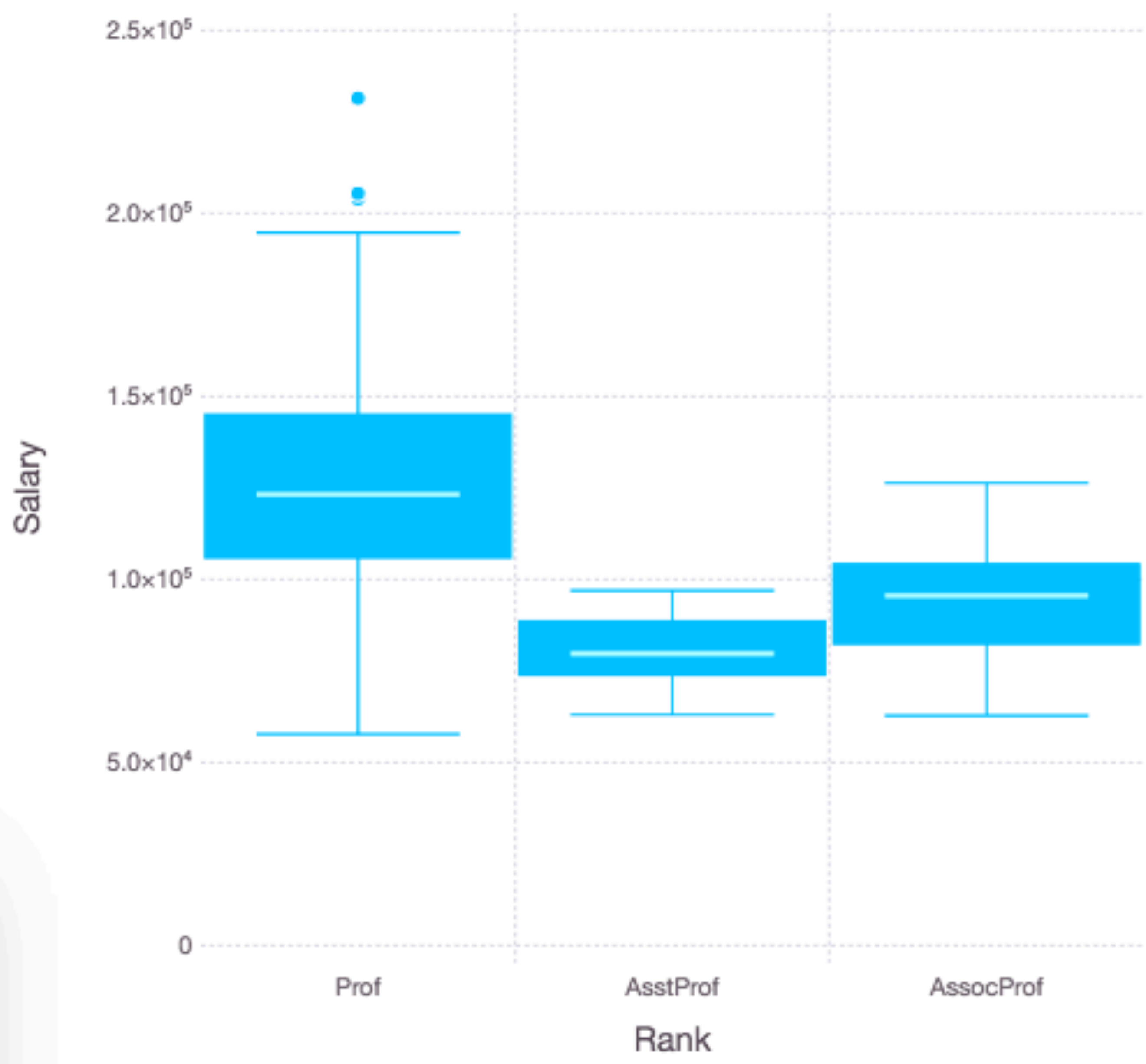


A = Theoretical

B = Applied

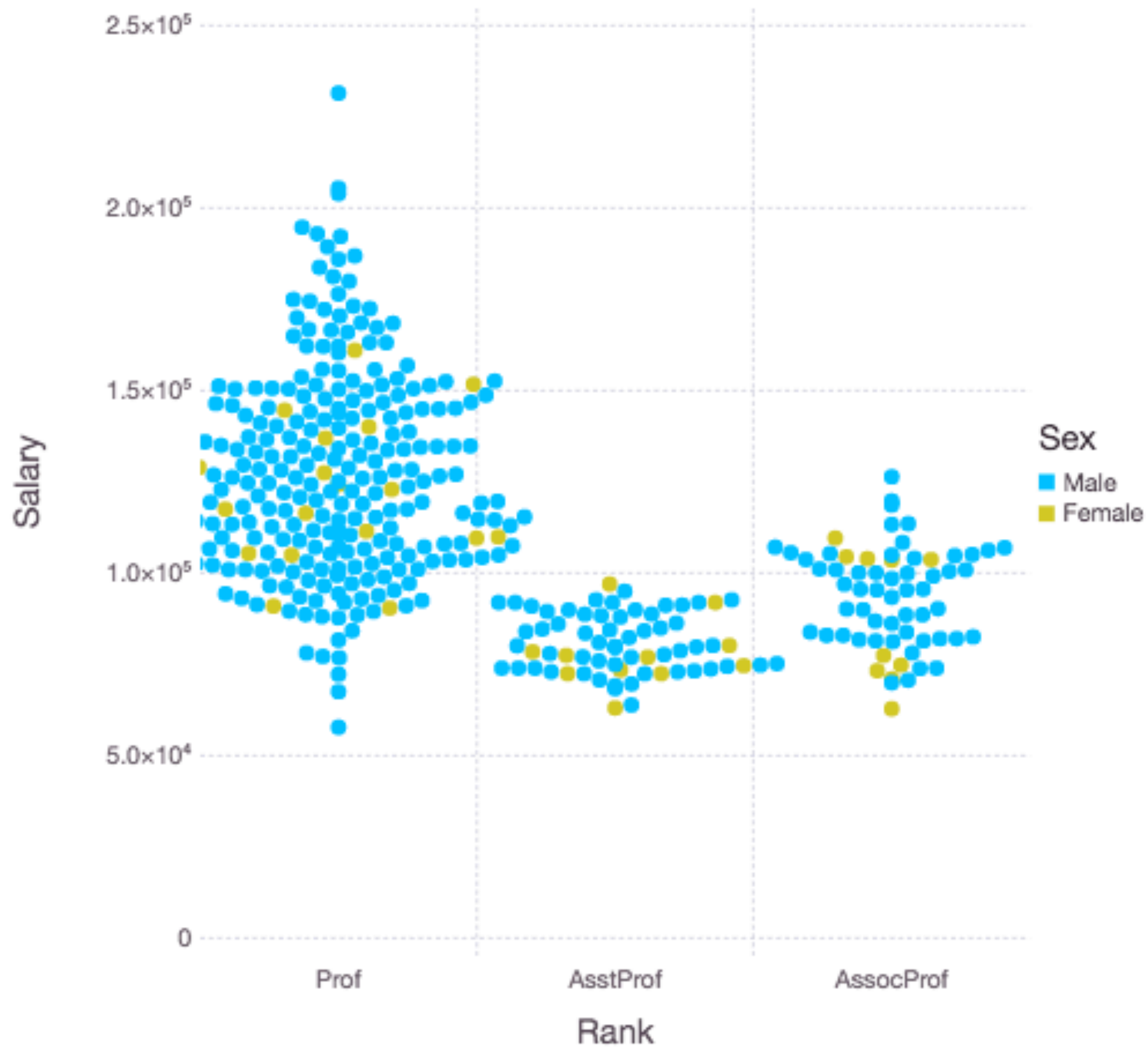
Salary by Rank

```
plot(dataset("car", "Salaries"), y="Salary", x="Rank",Geom.boxplot)
```



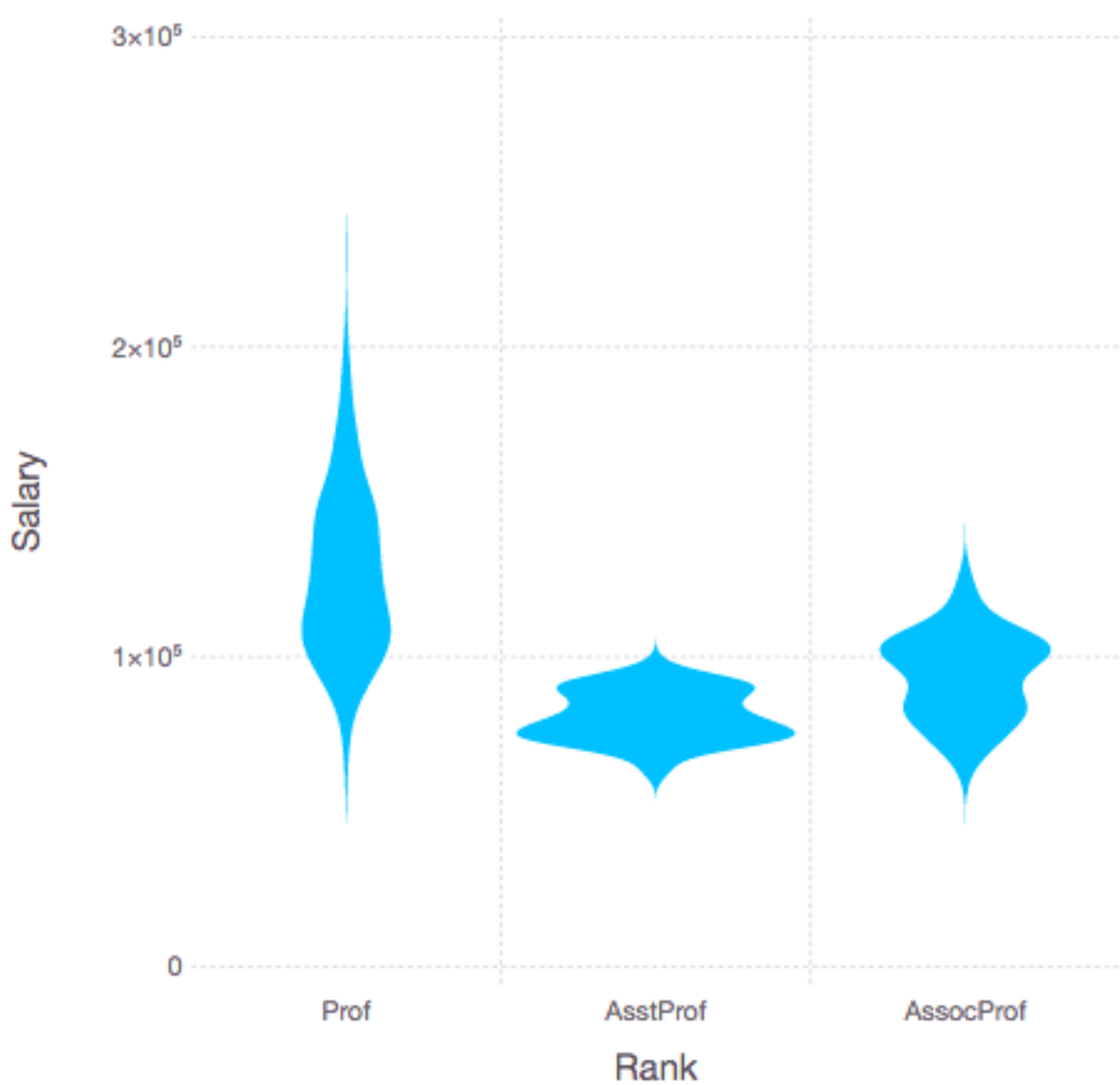
Beeswarm: Salary by Rank with Sex

```
plot(dataset("car", "Salaries"), x="Rank", y="Salary",color="Sex",Geom.beeswarm)
```



Violin Plot: Salary by Rank

plot(dataset("car", "Salaries"), x="Rank", y="Salary",Geom.violin)



Distributions

Think in distributions not numbers

Poincare's Baker

- France late 1800's
- Bread hand made, regulated
- Variation in weight of bread
- Poincare suspected baker of cheating

Dwell Time & A/B Testing of Websites

- Dwell time - how long people spend on a web page

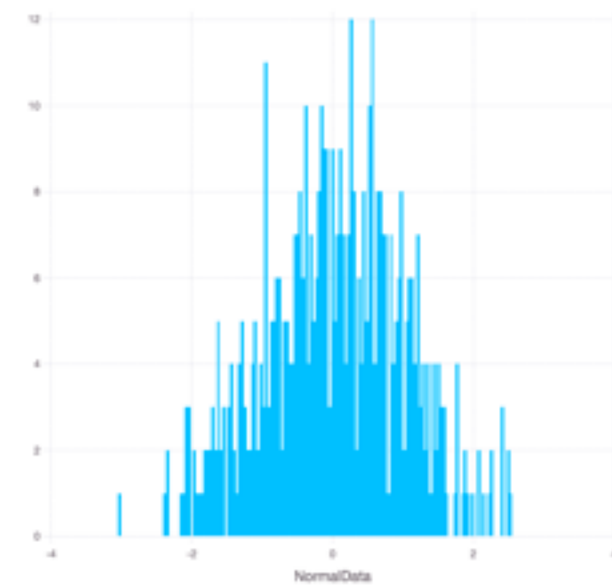
- A/B testing - Showing two versions of a page to different people

- How to tell if dwell time differs from between versions

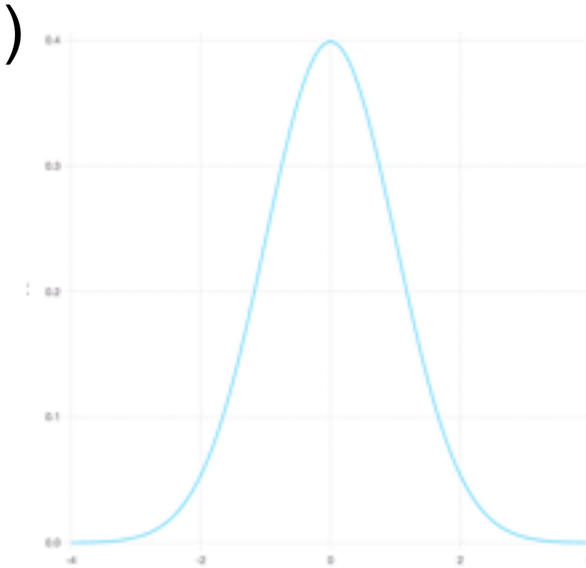
Distributions.jl

Generate common distributions
Fit data to distributions

```
using Gadfly
using DataFrames
using Distributions
```




```
normal_dist = Normal()
normal_sample = rand(normal_dist,500)
normal_dataframe = DataFrame(NormalData = normal_sample)
plot(normal_dataframe, x = "NormalData", Geom.histogram)
```

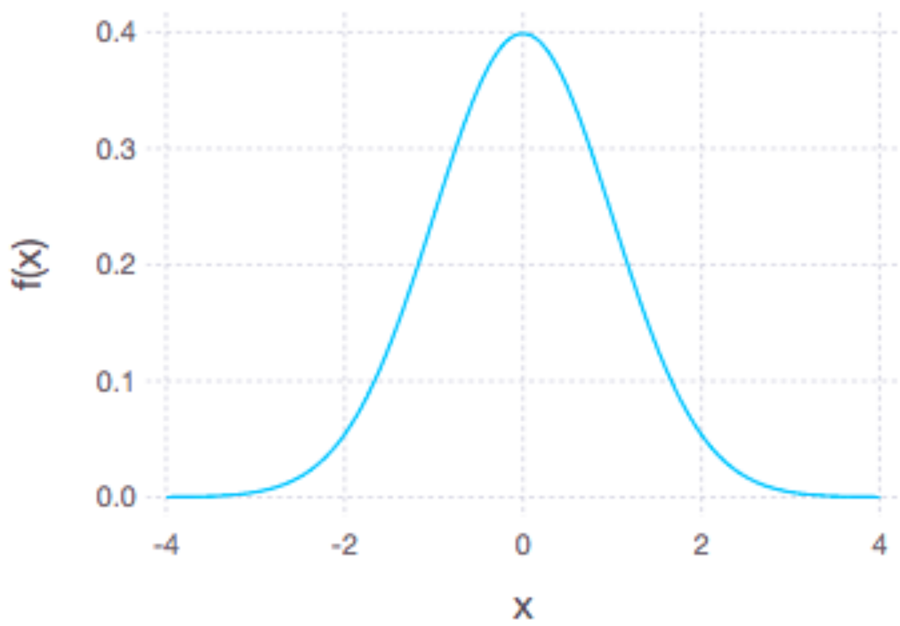


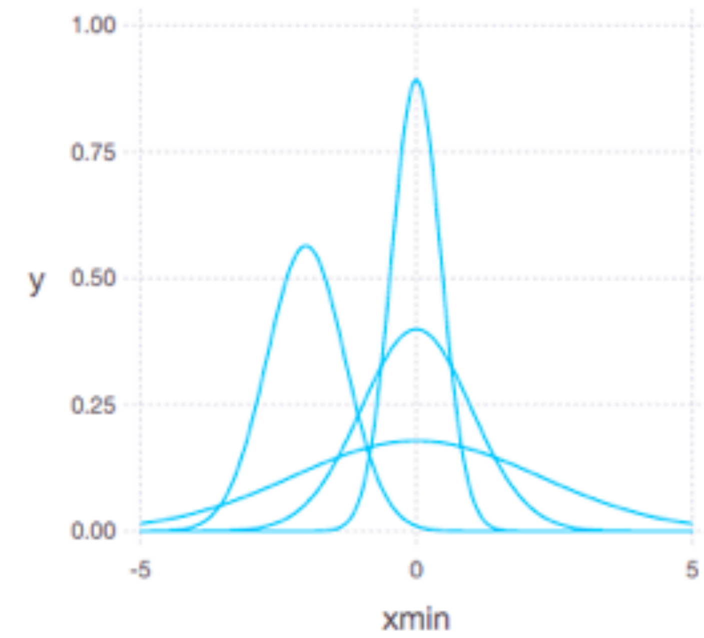
```
# pdf generates a function from the distribution
plot(x -> pdf(normal_dist,x), -4,4)
```

```
# fit
fitted_dist = fit(Normal,normal_sample)
```

Normal(μ=-0.0006388217034921672, σ=1.012334831313701)

Normal (Gaussian) Distribution



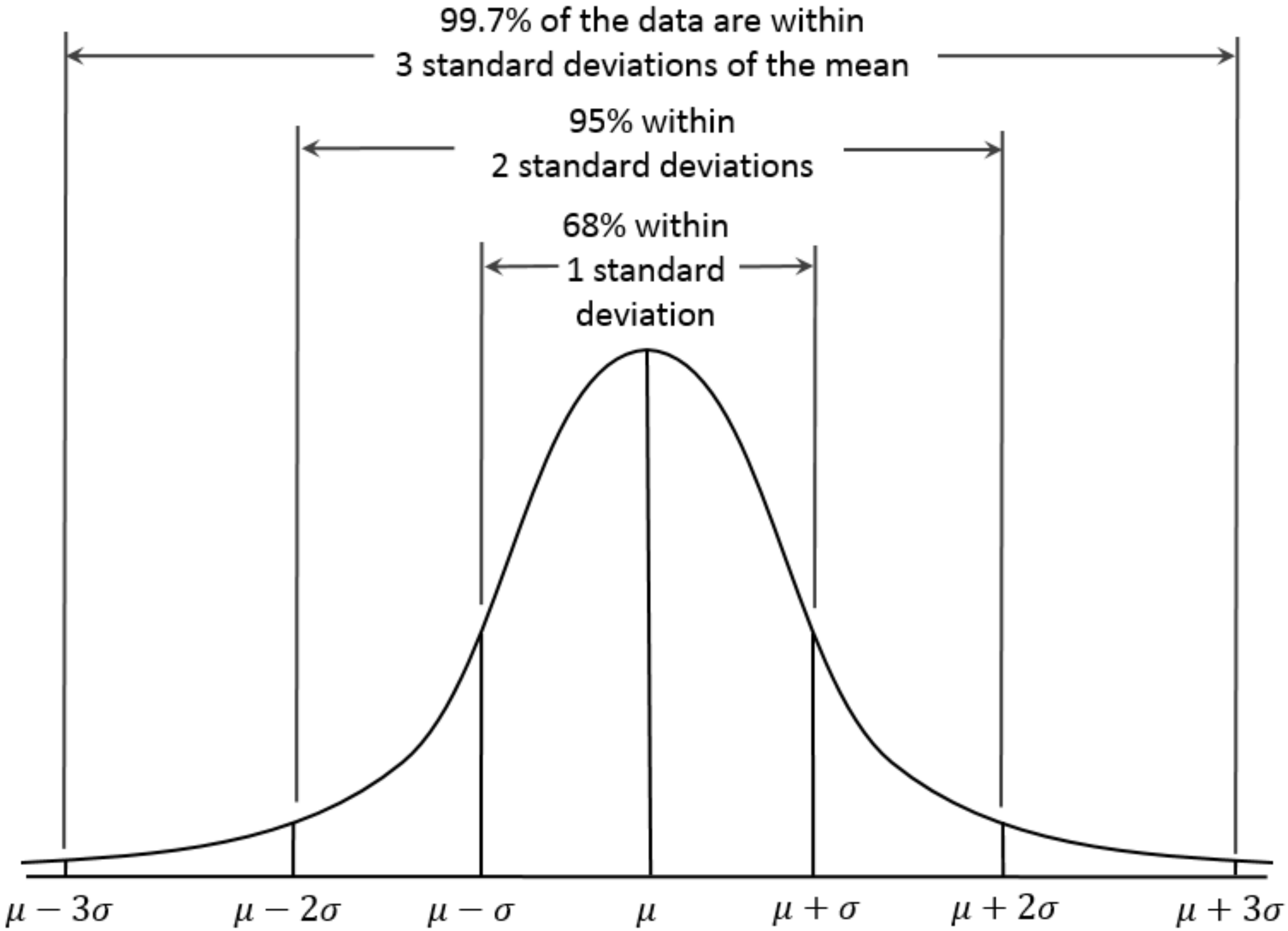


$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\sigma^2\pi}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Normal distribution is specified by

μ - mean, central point

σ - standard deviation



Monday, September 26, 16

Source: https://en.wikipedia.org/wiki/Normal_distribution

Populations & Samples

Measure	**Sample statistic**	**Population parameter**
Number of items	n	N
Mean	$\bar{x}$	μx
Standard deviation	S_x	σ_x
Standard error	$S_{\bar{x}}$	

Populations - all the items

Sample - set of representative items

Standard deviation of the sample-mean estimate of a population mean

Note to decrease the SE by 2 we need to increase the sample size by factor of 4

Hypothesis Testing

H_0 - Status quo
Null hypothesis

Poincare's Baker bread weight is correct

People spend the same amount of time on version A and B of the website

H_1 - What you are trying to prove
Alternative hypothesis

Poincare's Baker bread weight is less than it should be

People spend the more time on version A than B of the website

alpha - probability that H_1 is false

0.05
0.01
0.001

Sample N loaves of bread compute mean
If probability of that mean occuring from properly manufactured bread is less than 0.05 we accept H_1

Types of Errors

False Positive (FP), type I error

- Accepting H_1 when it is not true
- Smaller alpha values reduce FP

False Negative (FN), type II error

- Rejecting H_1 when it is true
- Small alphas increase FN

Causation & Correlation

Statistics

- Does not prove that one thing is caused by another
- Demonstrates that events are rare

If we accept H_1 with alpha = 0.05

- 5% chance that H_1 is wrong

If 100 studies accept H_1 with alpha = 0.05

- Expect about 5 of them are false positives

Sensitivity & Specificity

Sensitivity $$\frac{\text{Correctly predicted } H_1 \text{ cases}}{\text{Total number of } H_1 \text{ cases}}$$

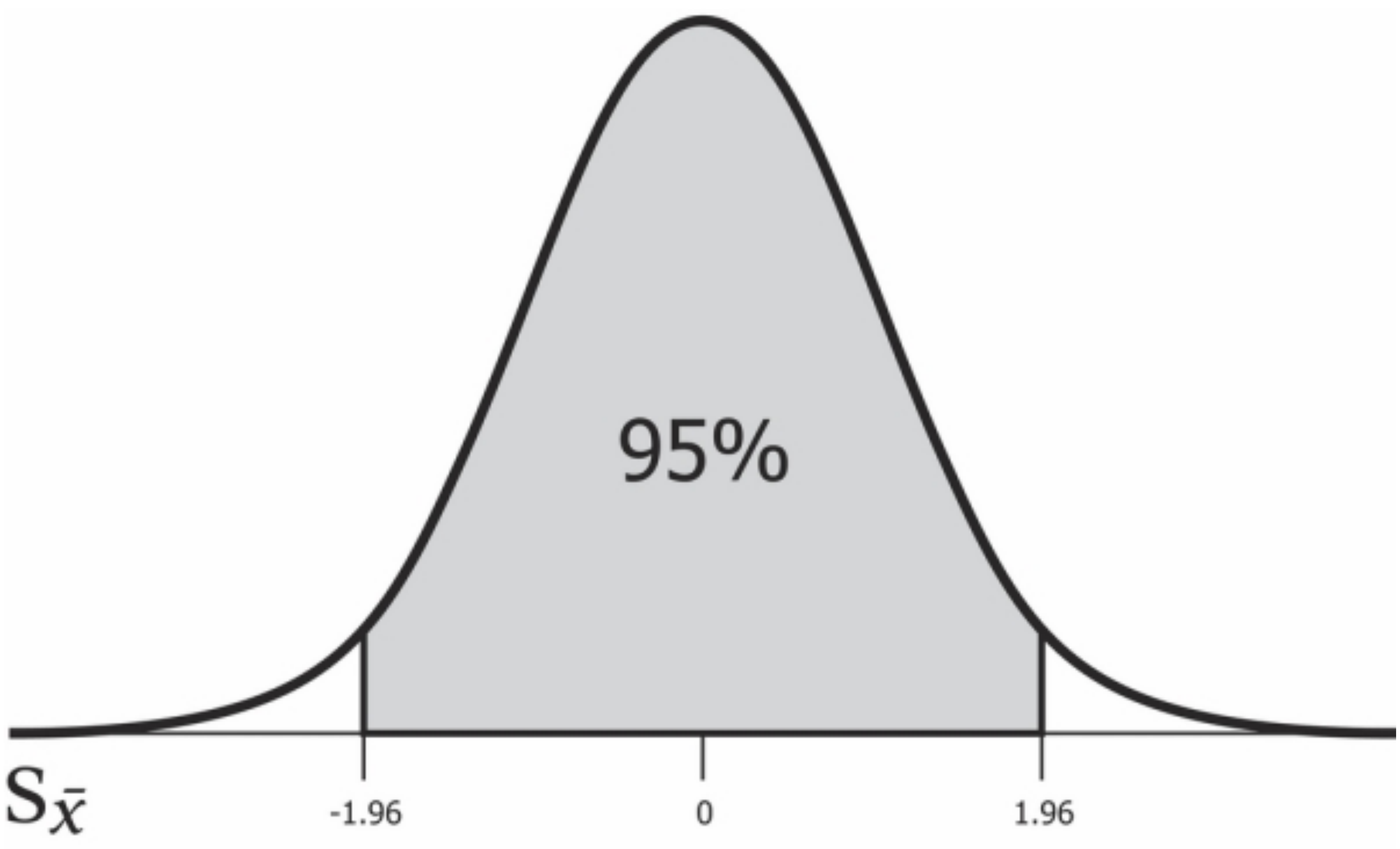
Specificity $$\frac{\text{Correctly predicted non-}H_1 \text{ cases}}{\text{Total number of non-}H_1 \text{ cases}}$$

Confidence Interval

Given a distribution and a p value

The interval that will contain 1-p of the values

95% Confidence, p = 0.05



±1.96*Standard Deviation

Computing Confidence Interval in Julia

```
using HypothesisTests

ci(OneSampleTTest(your_data))
ci(OneSampleTTest(your_data), 0.05)
```

OneSampleTTest

EqualVarianceTTest

Two samples come from a distributions with equal variances

UnequalVarianceTTest

Two samples come from a distributions with unequal variances

http://hypothesistestsjl.readthedocs.io/en/latest/parametric/test_t.html

Confidence Interval & Standard Error

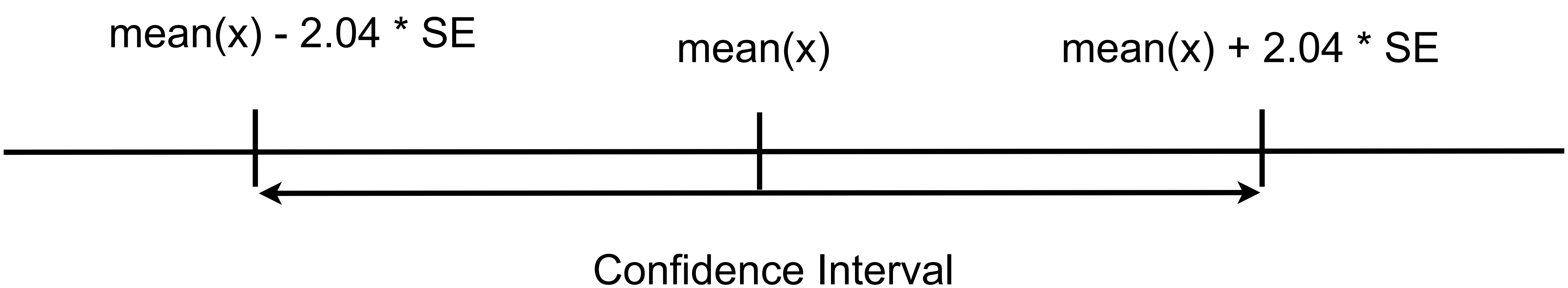
```
using Distributions
function t_test(x; conf_level=0.95)
    alpha = (1 - conf_level)
    tstar = quantile(TDist(length(x)-1), 1 - alpha/2)
    SE = std(x)/sqrt(length(x))

    lo, hi = mean(x) + [-1, 1] * tstar * SE
    "($lo, $hi)"
end
```

Sample Size 31

tstar = 2.04 alpha = 0.05
tstar = 2.75 alpha = 0.01
tstar = 3.65 alpha = 0.001

Sample Size 3000

tstar = 1.96 alpha = 0.05
tstar = 2.58 alpha = 0.01
tstar = 3.29 alpha = 0.001



Poincare's Baker

How to check for Cheating Bakers

Weigh N samples of bread

Compute confidence interval of the mean of the sample

See if expected mean is in confidence interval

Poincare's Baker

Assume

- Bread weight supposed to be 1000g
- Standard deviation of 30g
- Baker makes bread 20g lighter

```
using Distributions
using HypothesisTests

d = Normal(980,30)
fake_sample = rand(d,100)
(a,b) = ci(OneSampleTTest(fake_sample),0.01)
```

10 Samples

a	b
974.0	990.0
972.5	988.0
966.0	983.0
971.2	985.0
972.8	988.0
972.1	988.0
973.3	989.0
970.5	988.0
971.9	986.0
970.8	986.0

Poincare's Baker

Assume

- Bread weight supposed to be 1000g
- Standard deviation of 30g
- Baker makes bread 10g lighter

```
using Distributions
using HypothesisTests

d = Normal(990,30)
fake_sample = rand(d,100)
(a,b) = ci(OneSampleTTest(fake_sample),0.01)
```

10 Samples

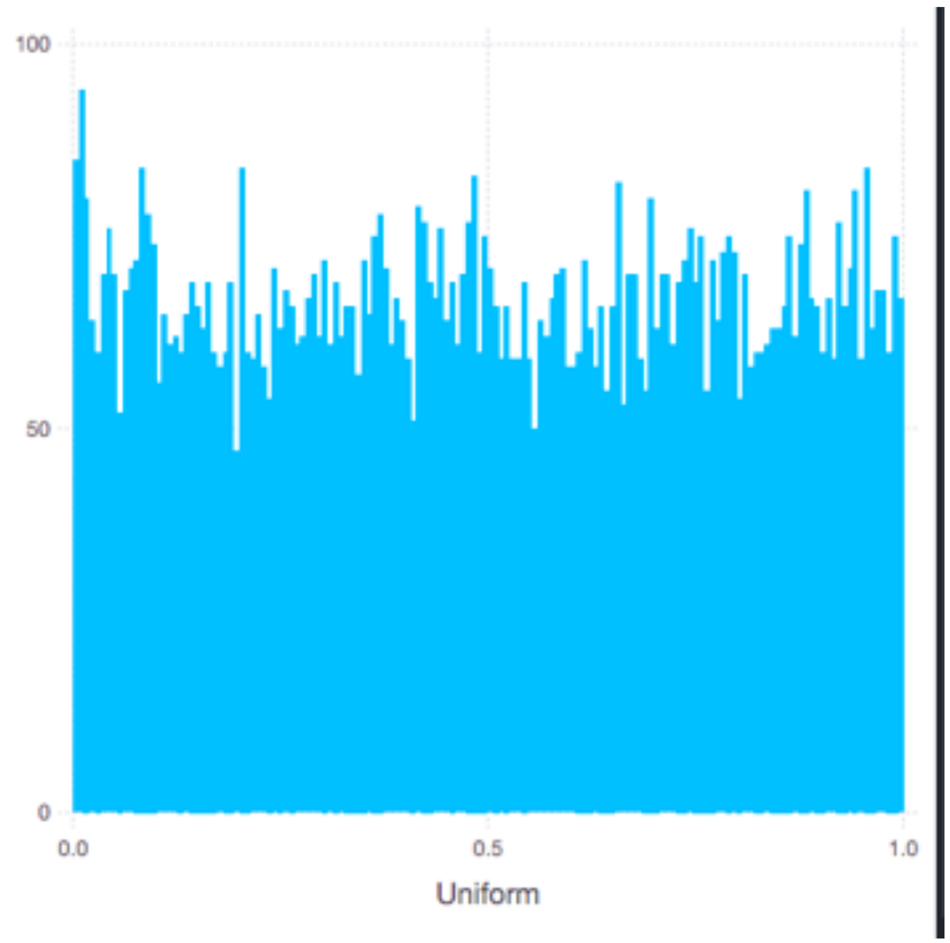
a	b
978.6	995.0
983.2	998.0
983.1	998.0
979.7	997.0
982.7	999.0
986.8	1000.0
983.7	999.0
979.9	995.0
981.3	997.0
984.8	1002.0

Central Limit Theorem

rand(n)

Generates n random numbers uniformly between 0 and 1

```
data = rand(10000)
plot(DataFrame(Uniform=data), x = "Uniform", Geom.histogram)
```



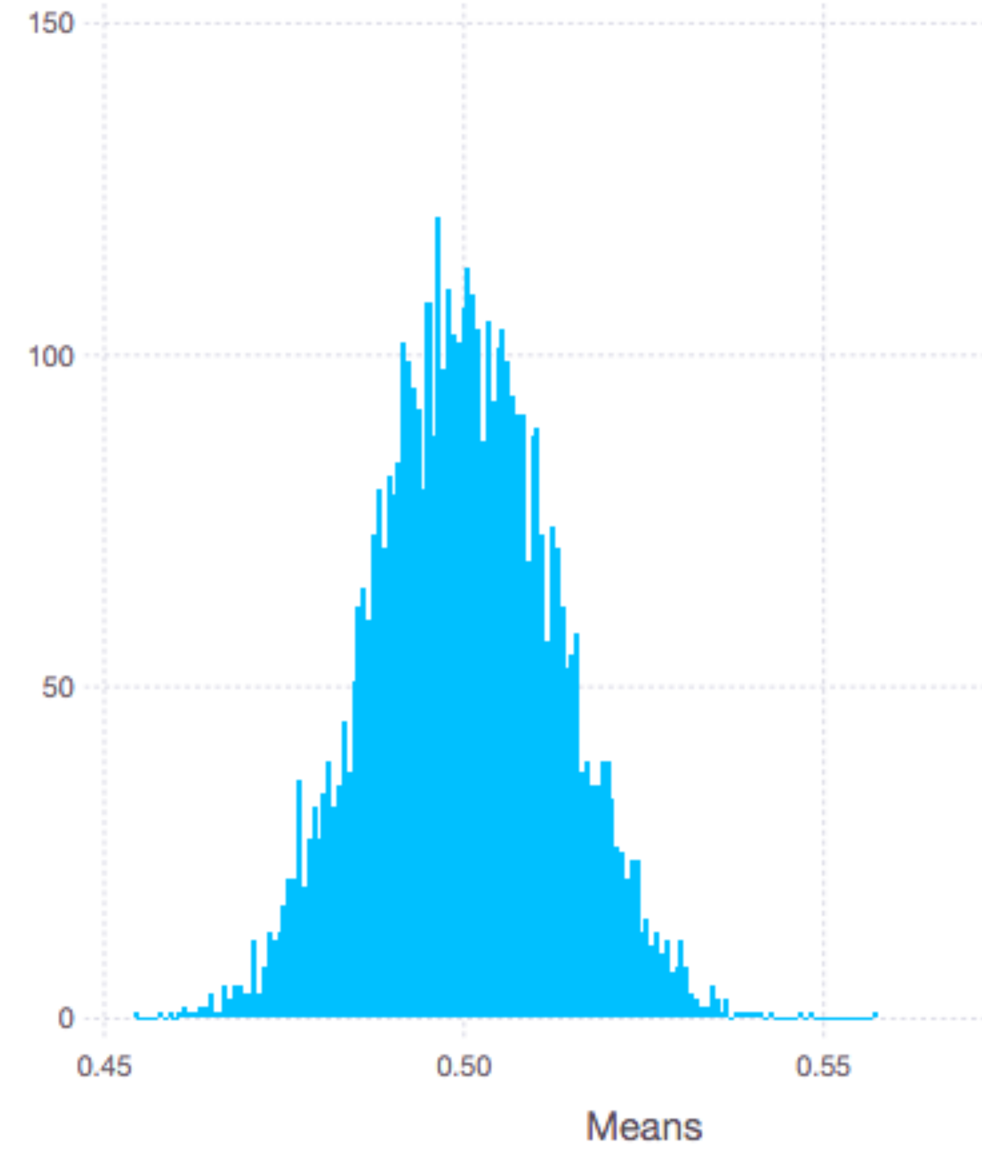
Central Limit Theorem

Let

$X_1, X_2, ..., X_N$ random sample

$S_N = (X_1 + ... + X_N)/N$

Then as N gets large S_N approximates the normal distribution




```
using Gadfly
using DataFrames
using Distributions

sample_mean(n) = sum(rand(n))/n
samples = map(x -> sample_mean(500),1:5000)
plot(DataFrame(Means= samples), x="Means", Geom.histogram)

fit(Normal,samples)

(μ=0.5000697736034079, σ=0.012822227485544065)
```

Dwell Times on Web sites

Look at Dwell data of website

Don't know the distribution of the dwell times

But daily mean of dwell times will be normally distributed

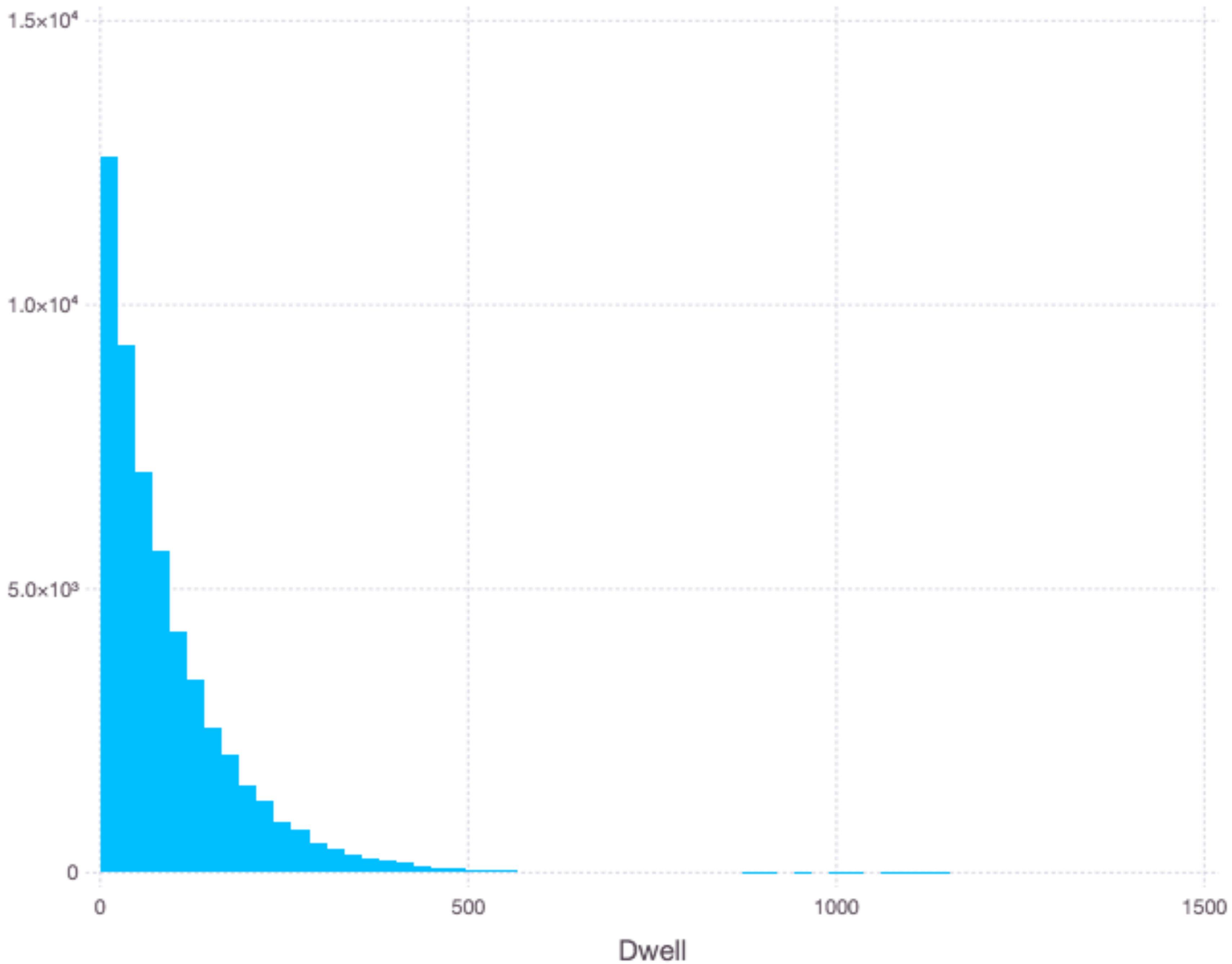
Dwell Data

```
data_location = "Some location on my hard drive"

dwell_times = readtable(data_location * "dwell-times.tsv", separator = '\t')
rename!(dwell_times,:dwell_time,:Dwell)
show(dwell_times)
```

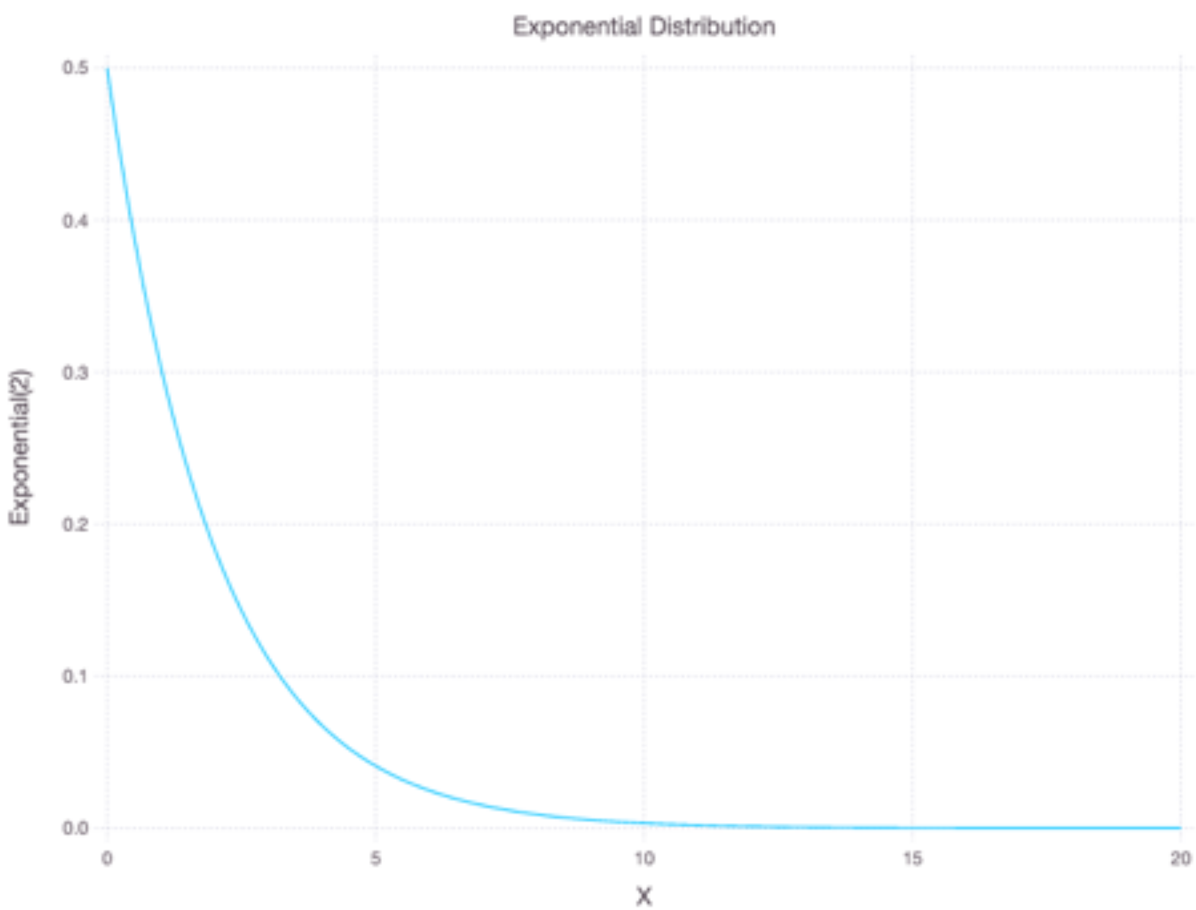
```
54000×2 DataFrames.DataFrame
| Row   | date                   | Dwell |
├───────┼────────────────────────┼───────┤
| 1     | "2015-01-01T00:03:43Z" | 74    |
| 2     | "2015-01-01T00:32:12Z" | 109   |
| 3     | "2015-01-01T01:52:18Z" | 88    |
| 4     | "2015-01-01T01:54:30Z" | 17    |
```

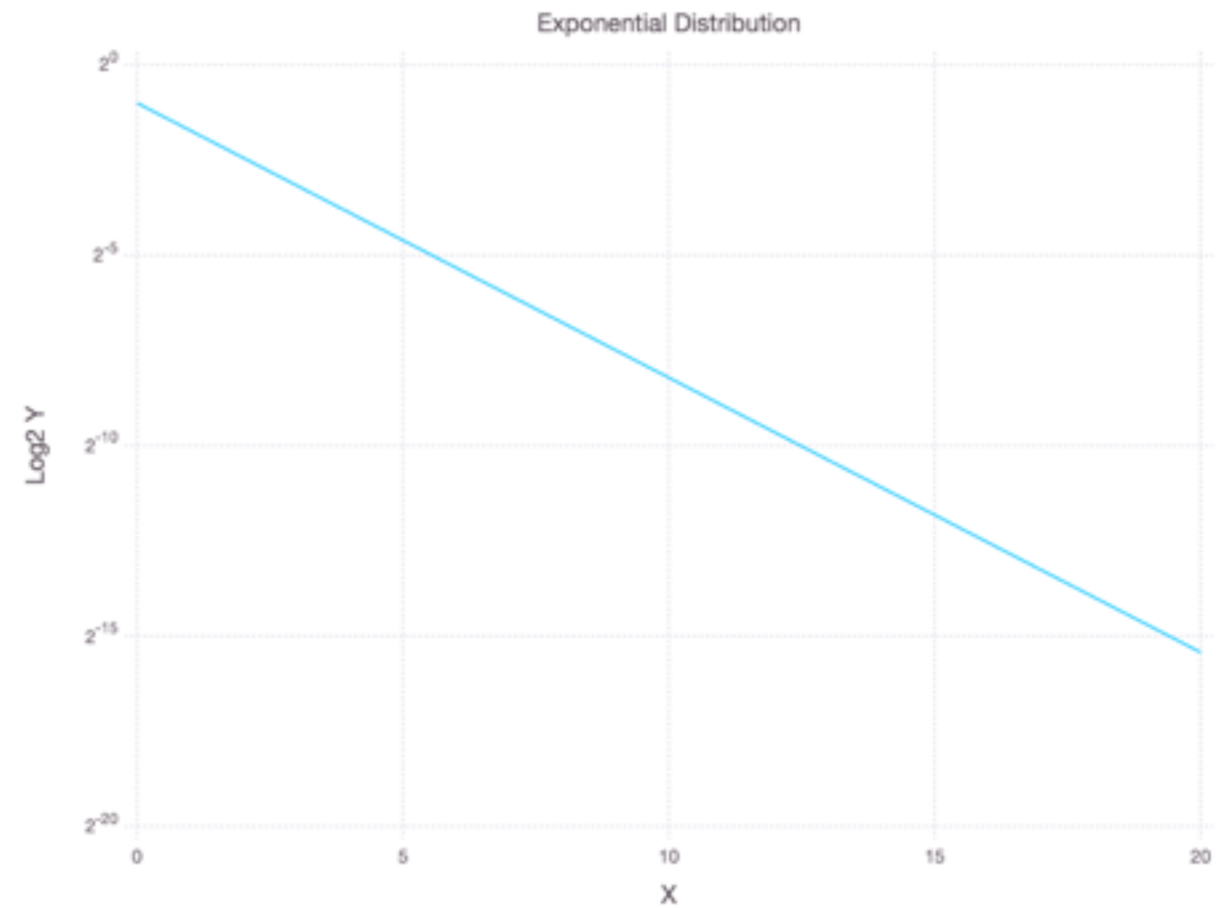
Dwell Times

plot(dwell_times, x="Dwell", Geom.histogram(bincount = 50))



Exponential Distribution

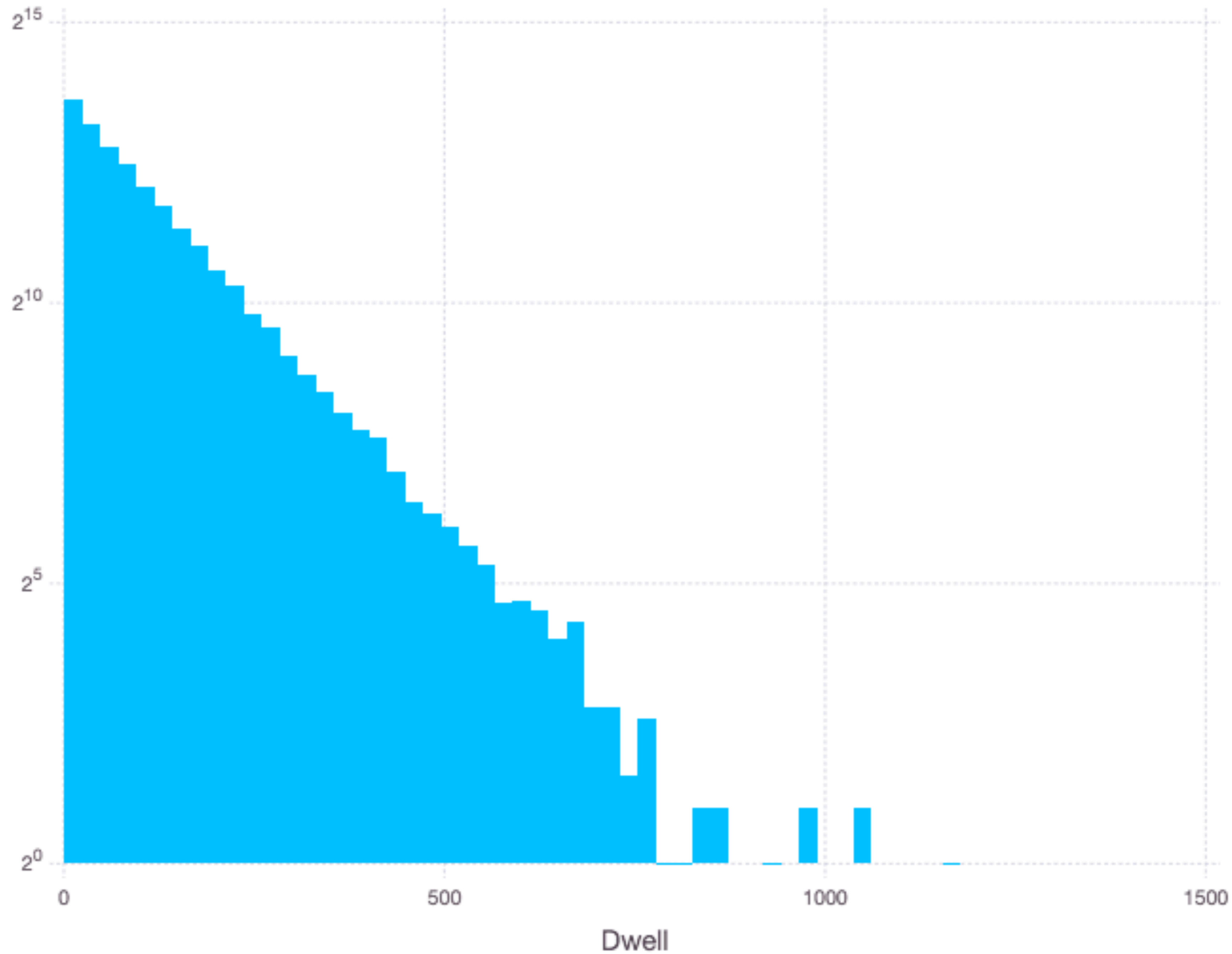




Log2(Y)

Log Scale - So Dwell Time is Exponential Dist.

```
plot(dwell_times, x="Dwell", Geom.histogram(bincount = 50), Scale.y_log2)
```



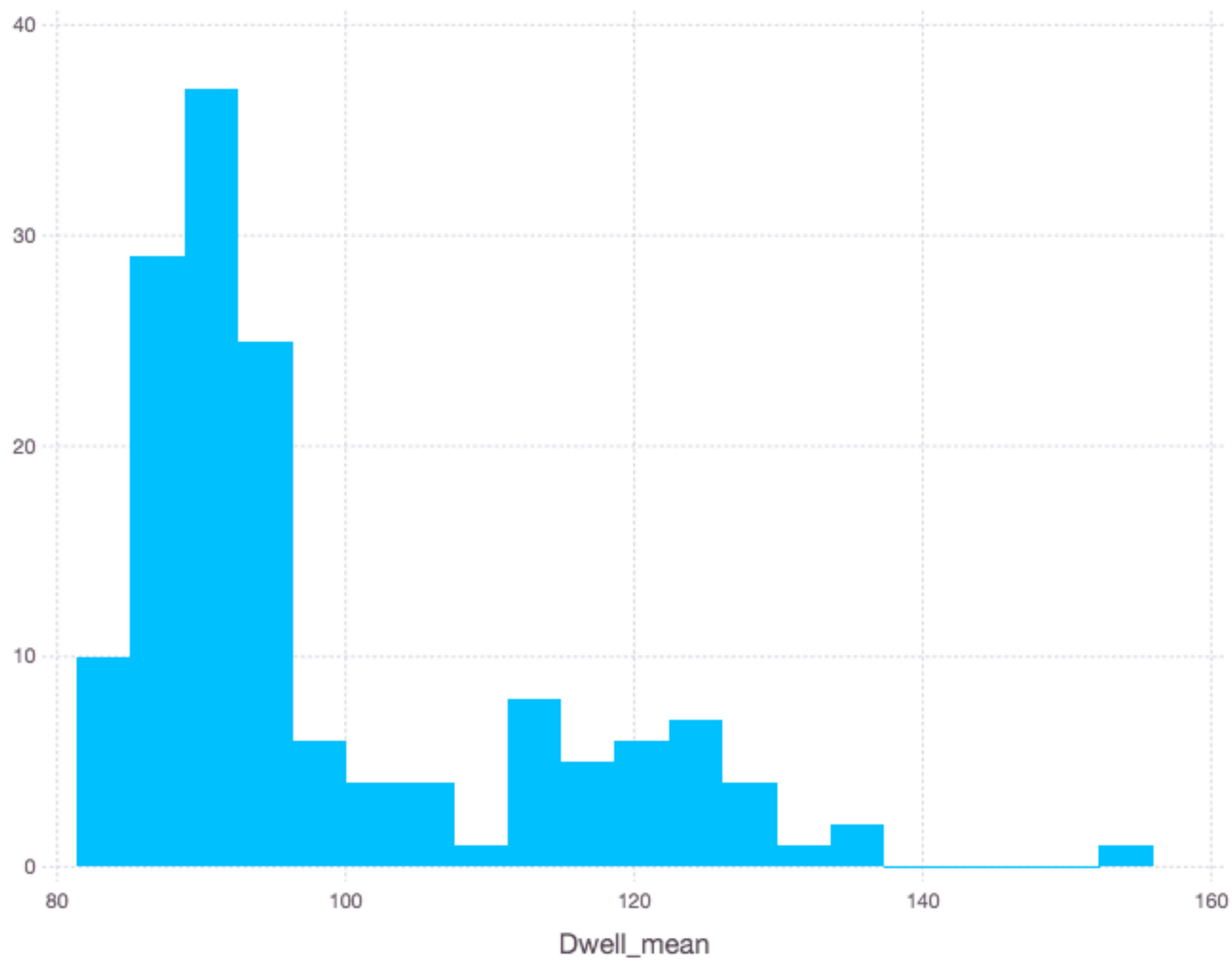
Compute Daily Mean

To use aggregate on date - so need to remove time from

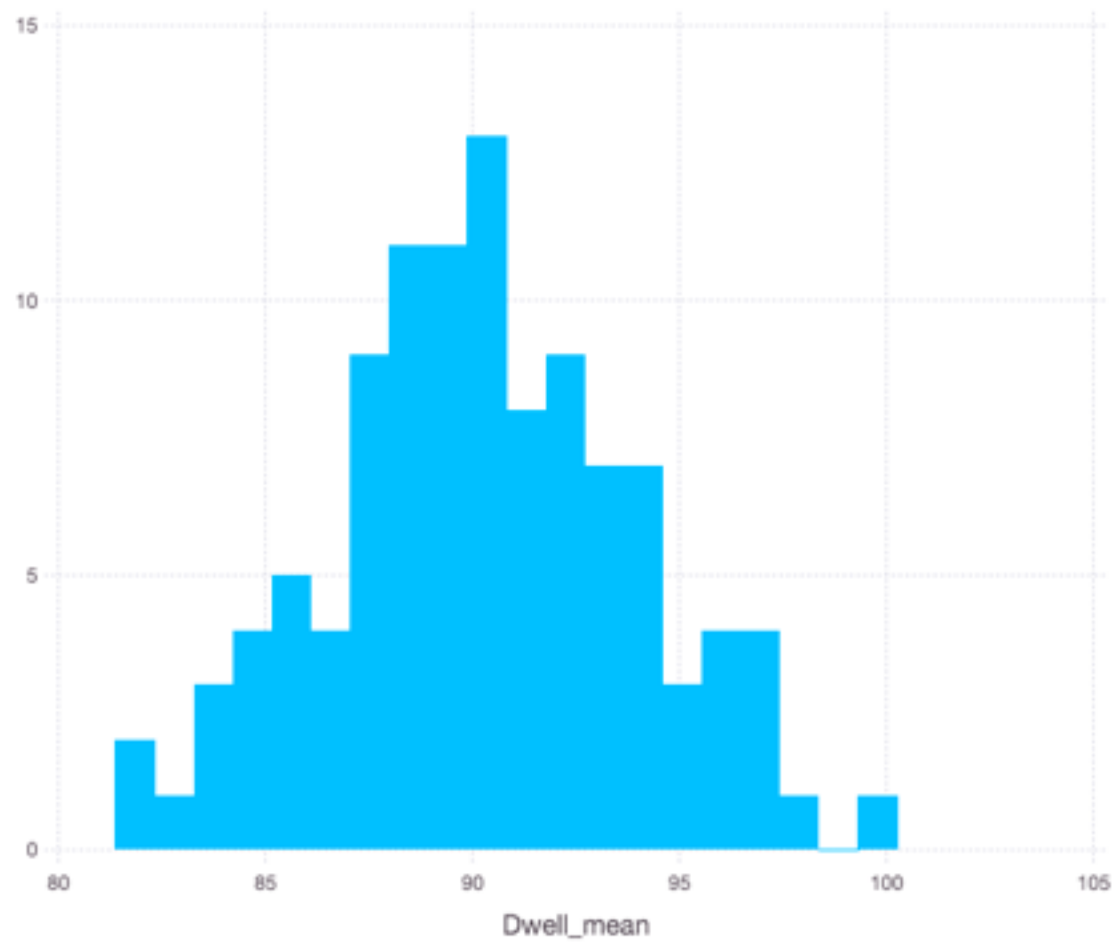
```
remove_time(s::String) = s[1:10]

function remove_time(d::DataFrame)
  d_copy = copy(d)
  rows = size(d)[1]
  for row in 1:rows
    d_copy[row,1] = remove_time(d[row,1])
  end
  d_copy
end

without_time = remove_time(dwell_times)

daily_dwell = aggregate(without_time,:date, mean)
```

Central Limit Theorem

plot(daily_dwell, x="Dwell_mean", Geom.histogram(bincount=20))



Week Days



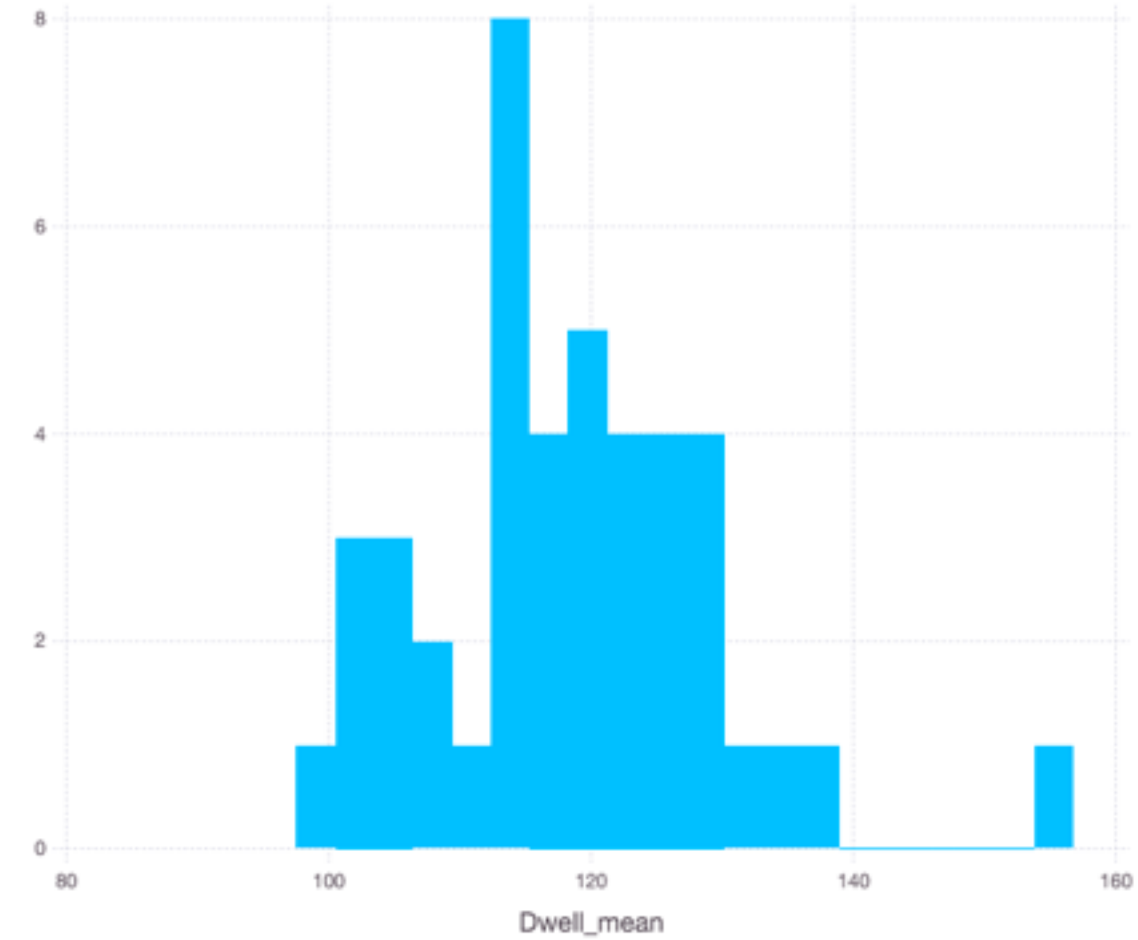
sample size = 107

mean = 90.2

std = 3.7

CI of mean p = 0.05 (115,122)

Weekends



sample size = 107

mean = 118.3

std = 11.0

CI of mean p = 0.05 (89.5 ,90.9)

Pvalue

Probability that the two samples are taken from the same distribution

```
using HypothesisTests
pvalue(UnequalVarianceTTest(weekend[:Dwell_mean],week_day[:Dwell_mean]))
```

8.25e-21