CS 649 Big Data: Tools and Methods
Fall Semester, 2021
Doc 13 Statistics, Sampling, Bloom
Feb 17, 2022

Descriptive Statistics

mean
median
mode
variance
standard variation
quantiles

Descriptive Statistics

Arithmetic mean

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

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

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

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

Standard Deviation, (SD, s, σ)

square root of the variance

Bessel's Correction

Normally only have a sample of data

Computing mean from sample introduces bias

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

Bessel's correction for this bias

Divide by N-1

For large N this is not needed

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

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

Python functions

Use Bessel's correction

```
data = pd.Series([2,4,4,4,5,5,7,9])

data.var()
data.std()
data.mean()
data.median()

data.skew()            0.8184875533567997

pd.Series([1,2,3,4,5,10,20,50,100,1000]).skew()
```

PySpark

```
from pyspark.sql import SparkSession
import numpy as np
import pandas as ps
spark = SparkSession.builder.getOrCreate()
```

```
pdf = ps.DataFrame({'A': np.random.rand(500)})
psdf = spark.createDataFrame(pdf)
```

```
import pyspark.sql.functions as F
result_df = (
    psdf
    .select(F.mean('A').alias('mean'),
            F.stddev('A').alias('stddev'),
            F.var_pop('A'),
            F.var_samp('A').alias('variance'))
)
result_df.show()
```

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

1 1 4 | 7 7 8 | 10 15 17 | 17 25 26

first quartile
Q1

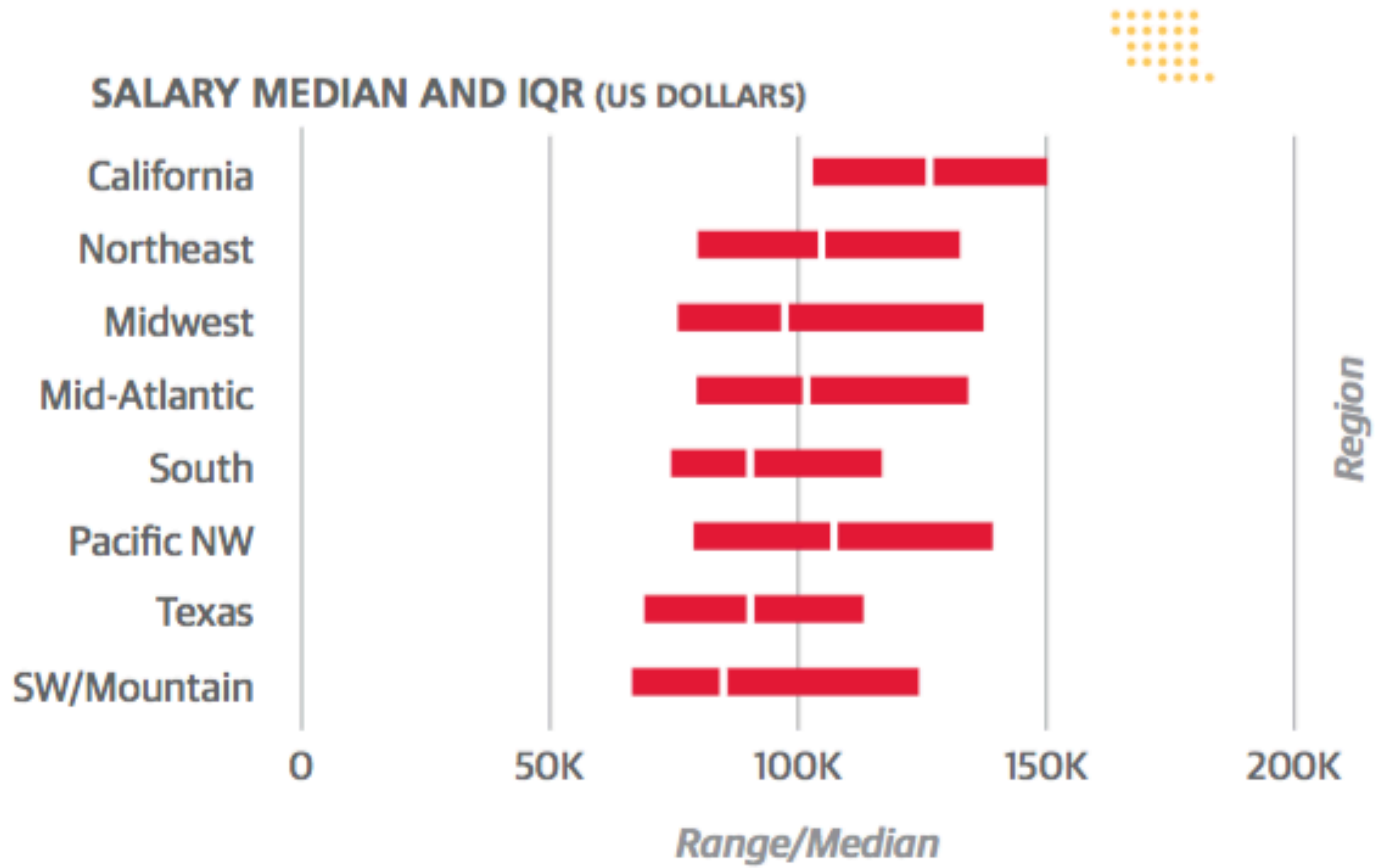
second quartile
median
Q2

third quartile
Q3

```
pd.Series([1, 1, 4, 7, 7, 8, 10, 15, 17, 17, 25, 26]).quantile([0.25, 0.5, 0.75])

0.25     6.25
0.50     9.00
0.75    17.00
dtype: float64
```

Red Bar shows middle two quartiles

White bar is median



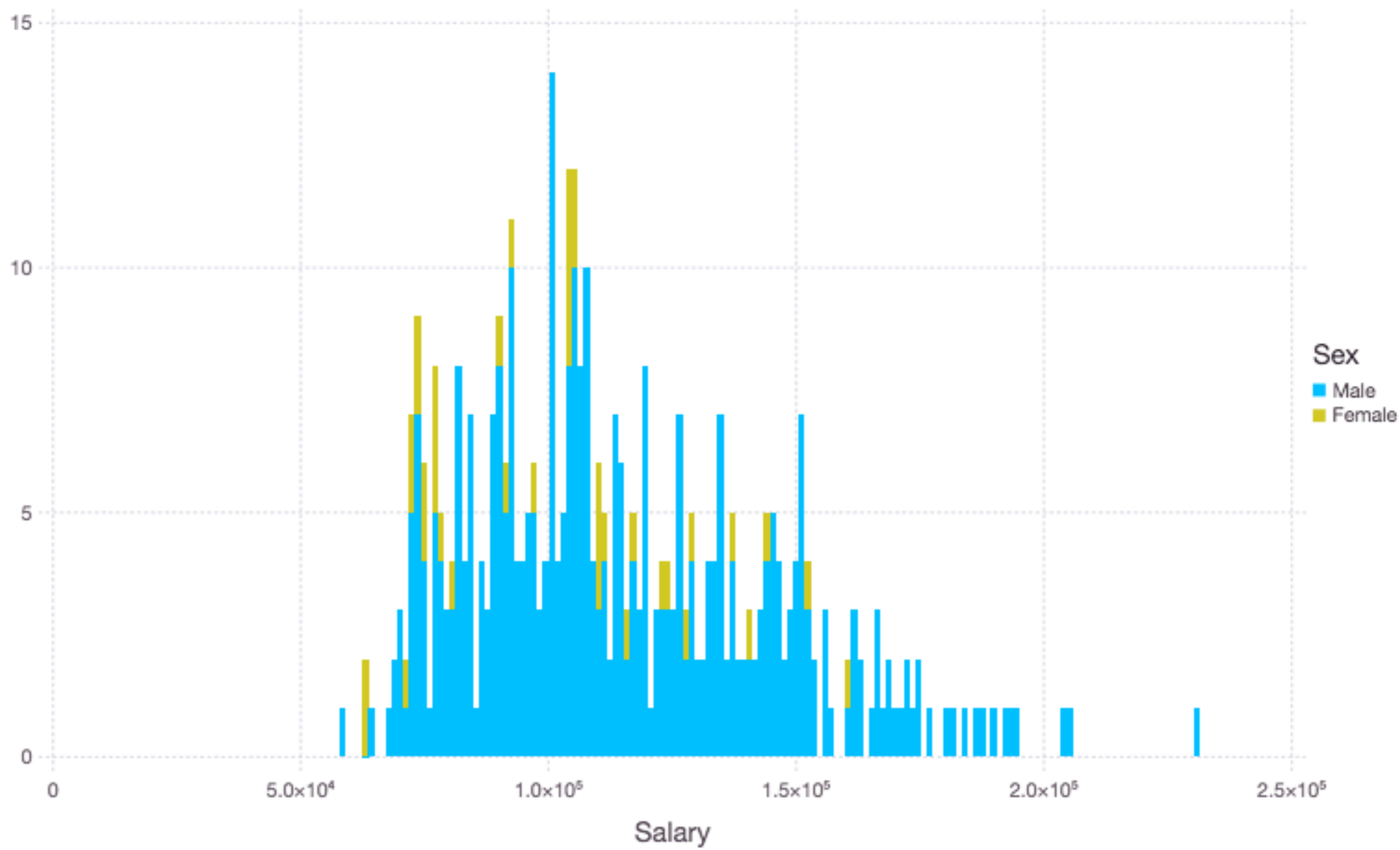
2008-9 Academic Salary

```
salaries_url = "https://vincentarelbundock.github.io/Rdatasets/csv/carData/Salaries.csv"
salaries = pd.read_csv(salaries_url, index_col=0)
```

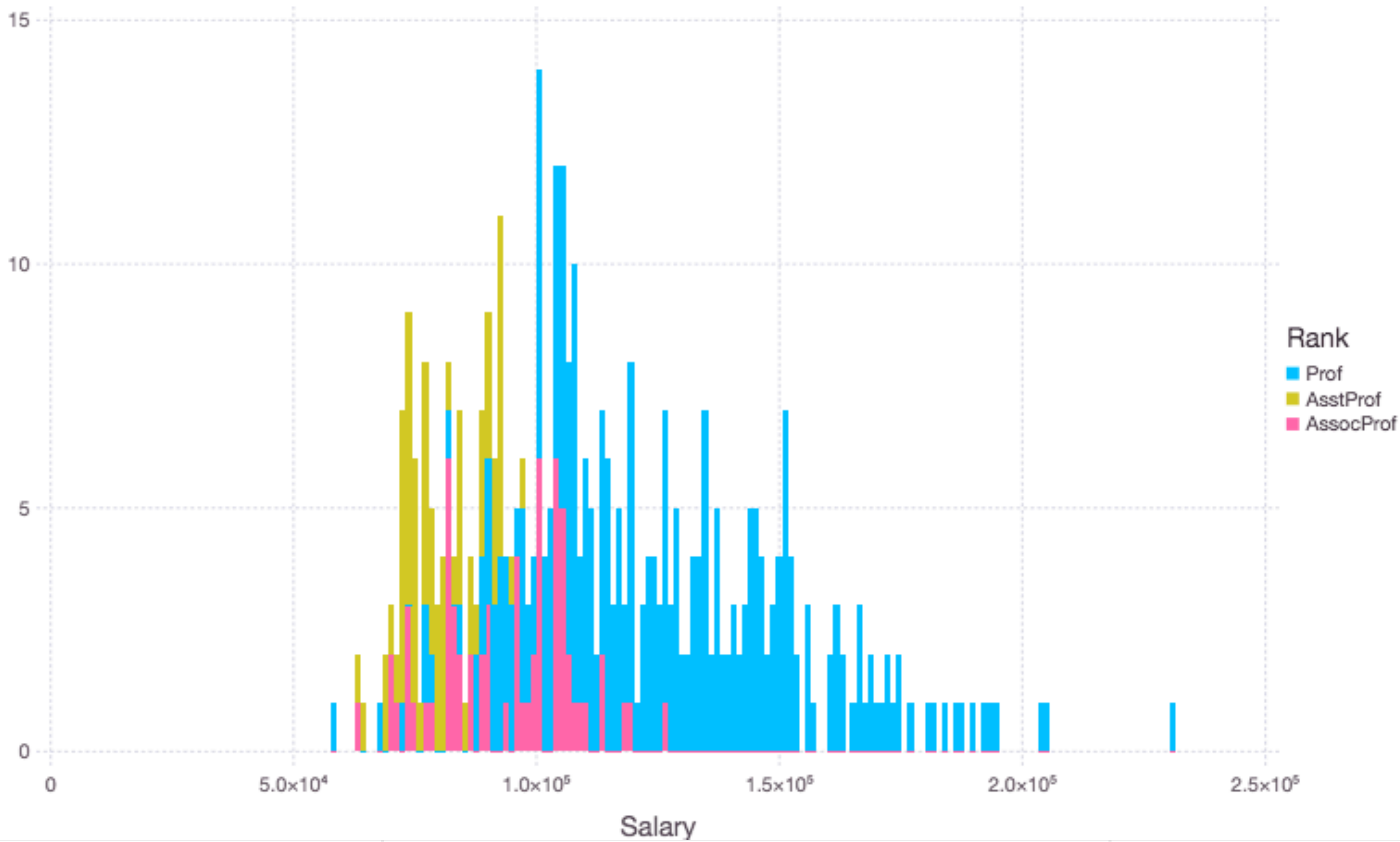
	rank	discipline	yrs.since.phd	yrs.service	sex	salary
1	Prof	B	19	18	Male	139750
2	Prof	B	20	16	Male	173200
3	AsstProf	B	4	3	Male	79750
4	Prof	B	45	39	Male	115000
5	Prof	B	40	41	Male	141500

A = Theoretical Department
B = Applied Department

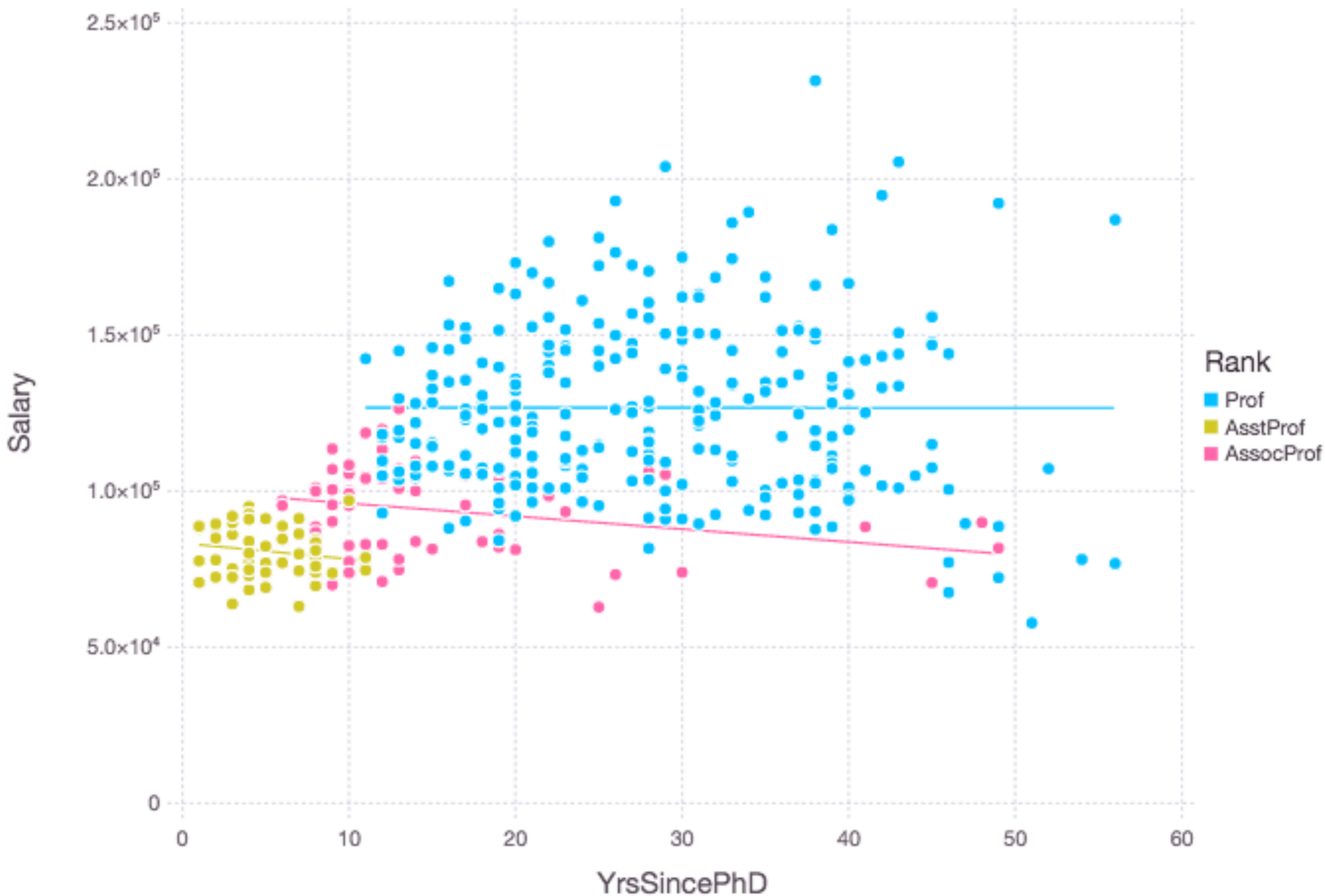
Salary & Sex


15
10
5
0
0
5.0×10⁴
1.0×10⁵
1.5×10⁵
2.0×10⁵
2.5×10⁵
Salary
Sex
Male
Female

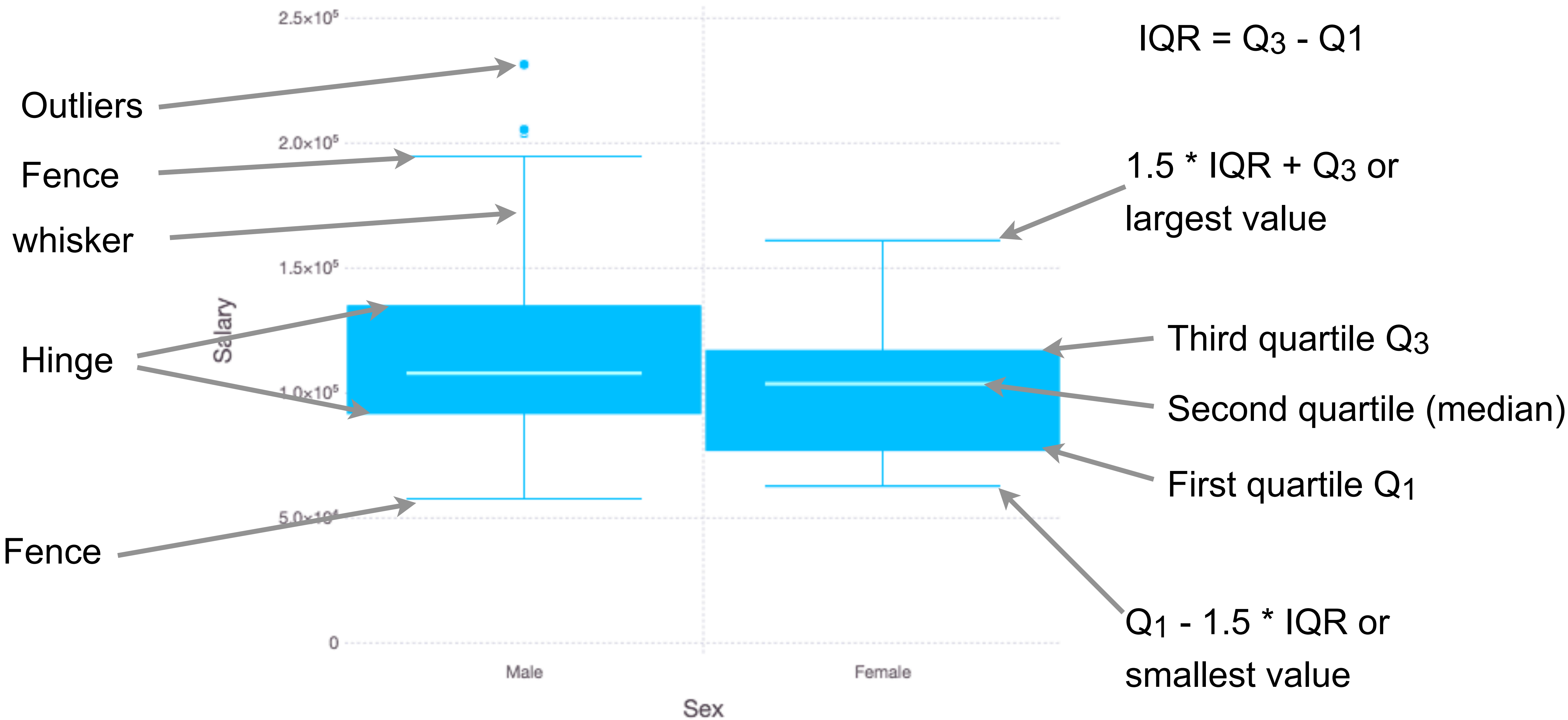
Salary & Rank


15
10
5
0
0
5.0x10⁴
1.0x10⁵
1.5x10⁵
2.0x10⁵
2.5x10⁵
Salary
Rank
Prof
AsstProf
AssocProf

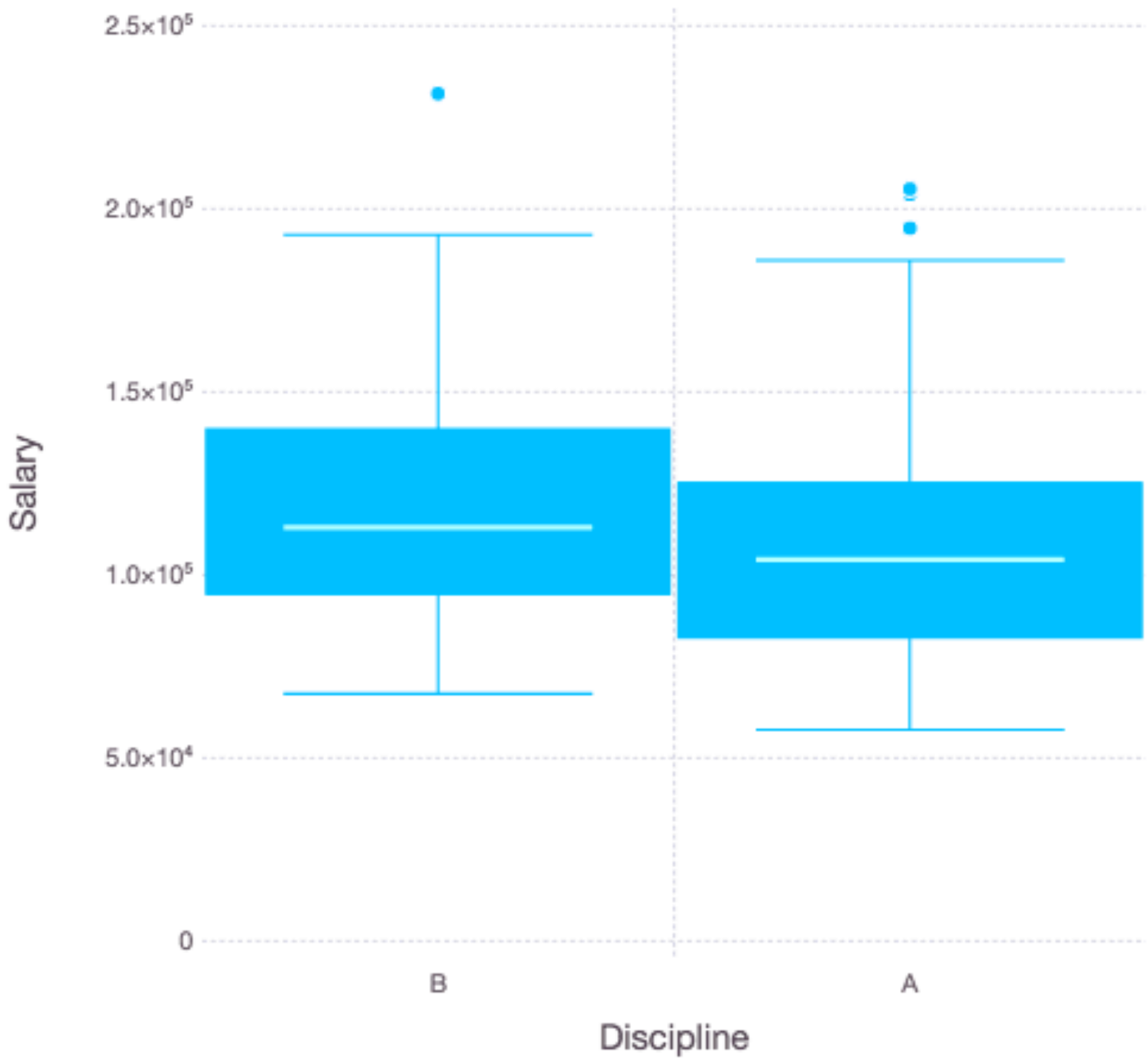
Scatter Plot: Salary-Years Colored by Rank



Box Plots (Tukey Method)



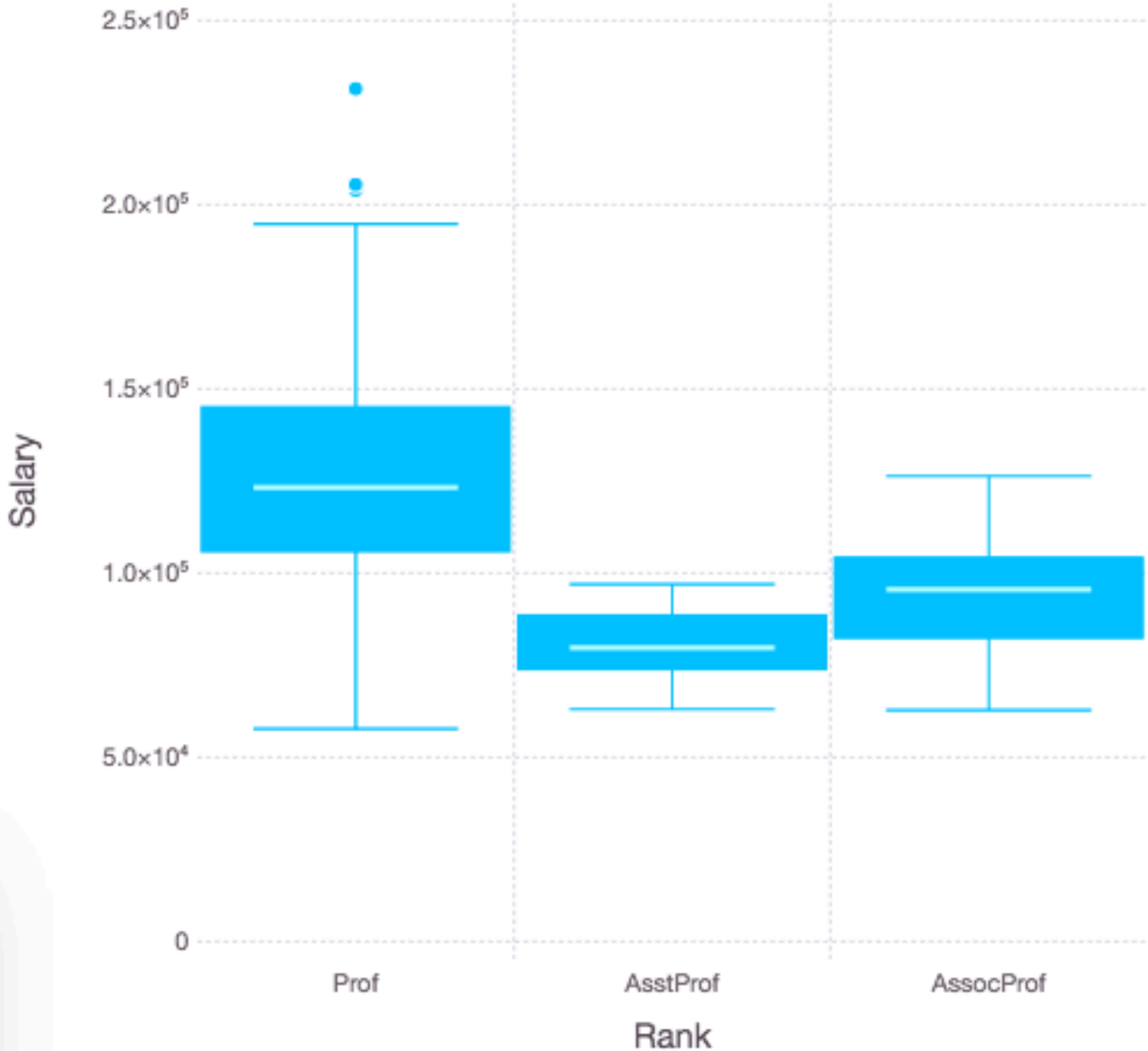
Salary by Discipline



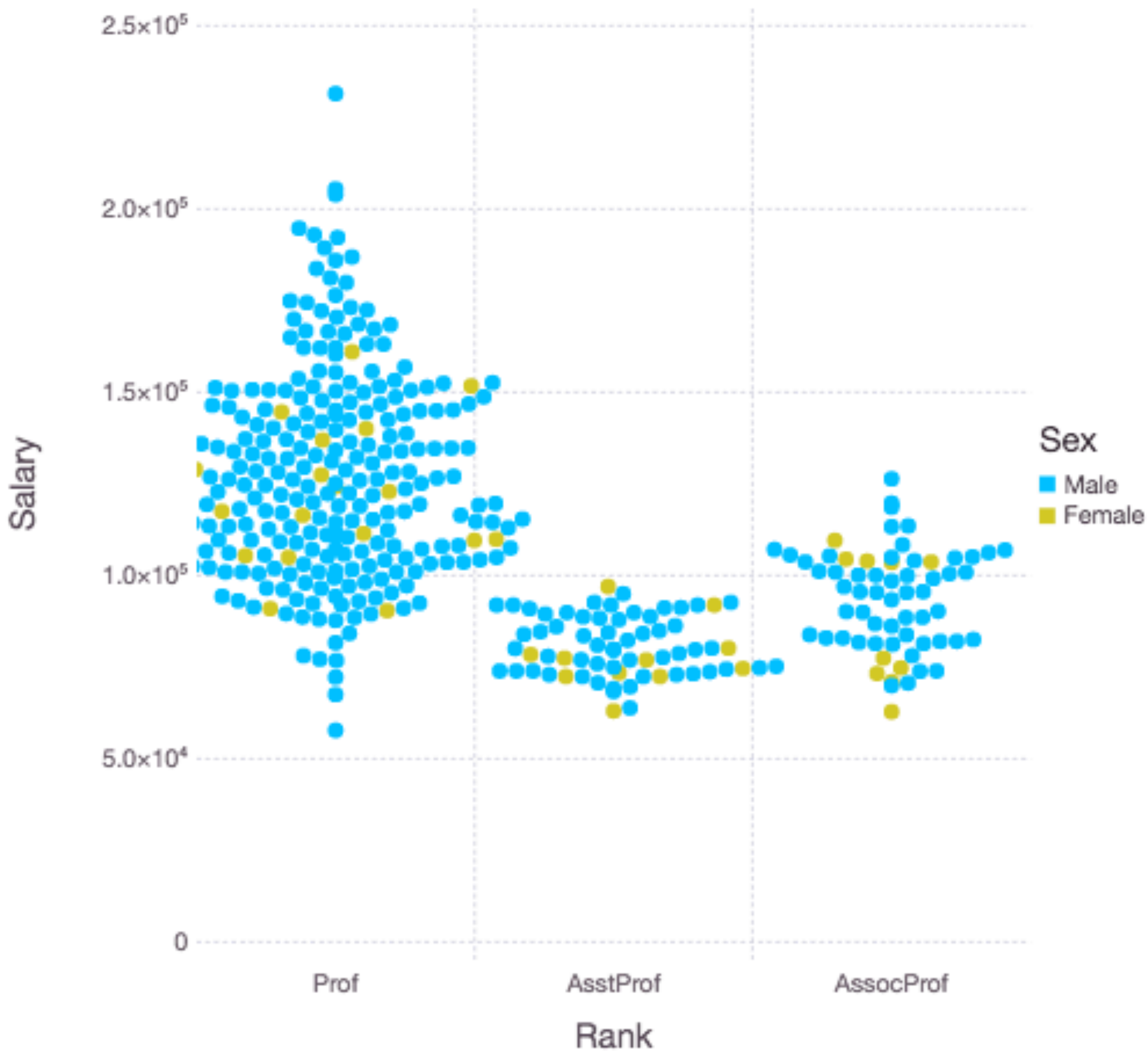
A = Theoretical

B = Applied

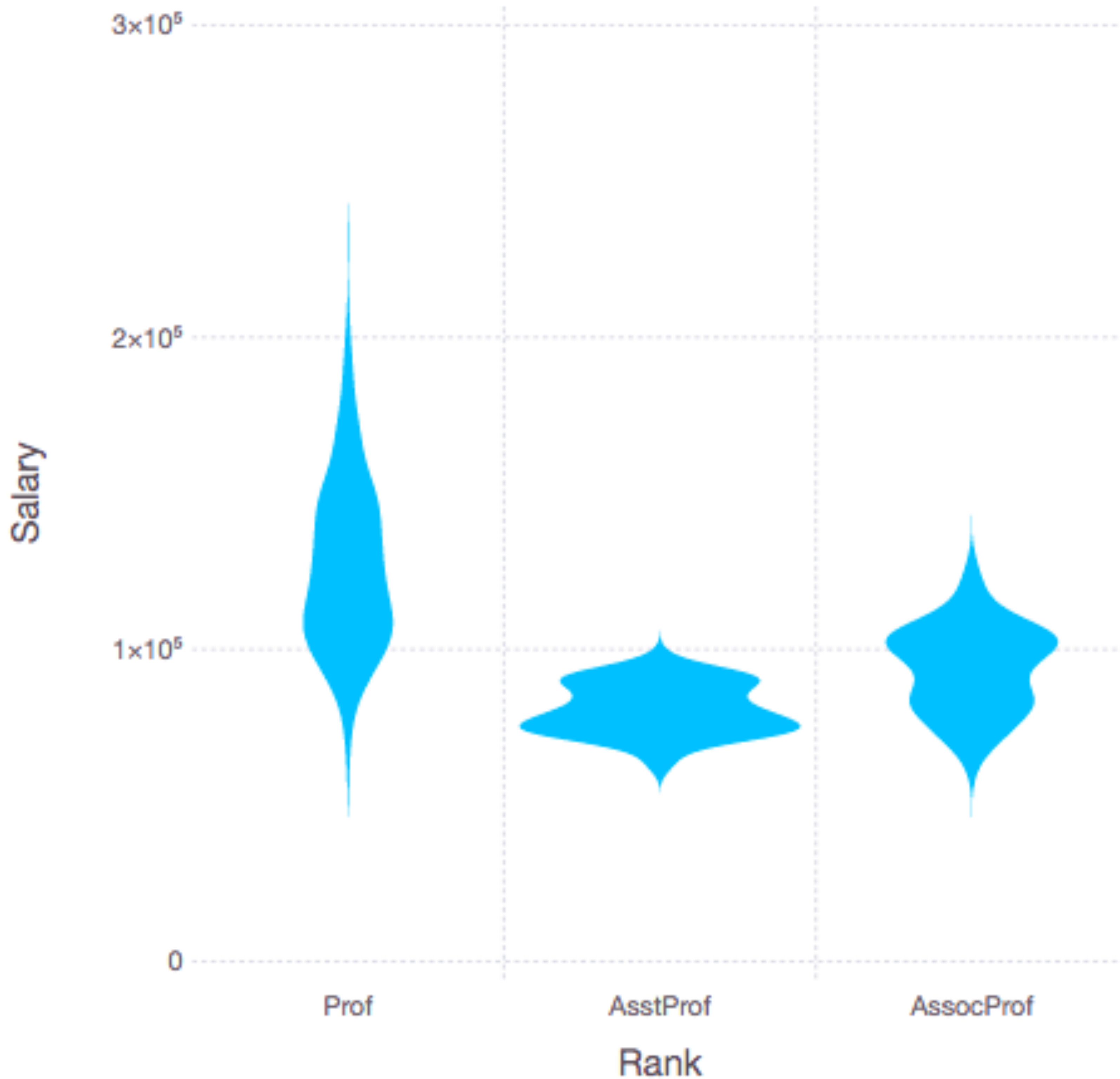
Salary by Rank



Beeswarm: Salary by Rank with Sex



Violin Plot: Salary by Rank



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

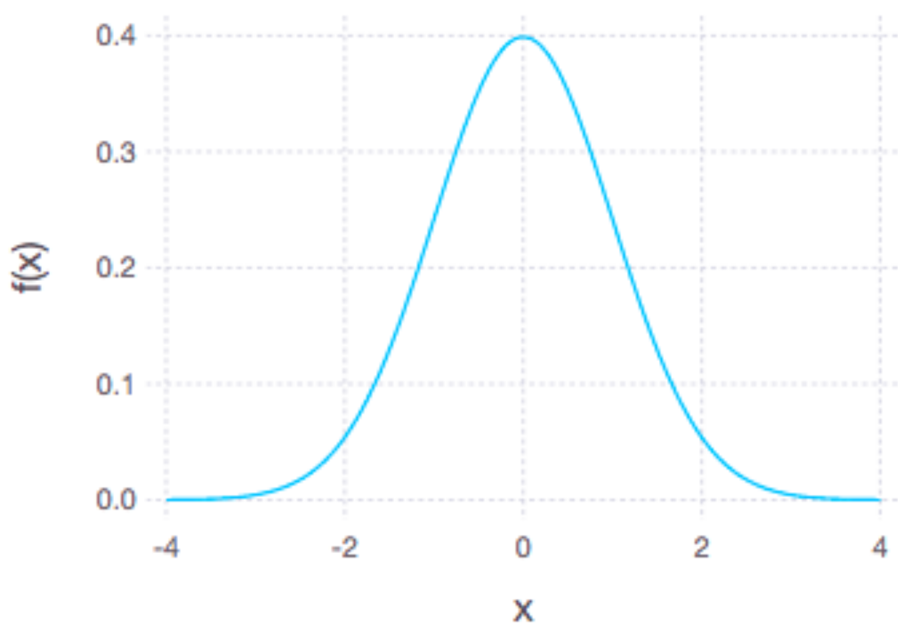
Dwell Time & A/B Testing of Websites

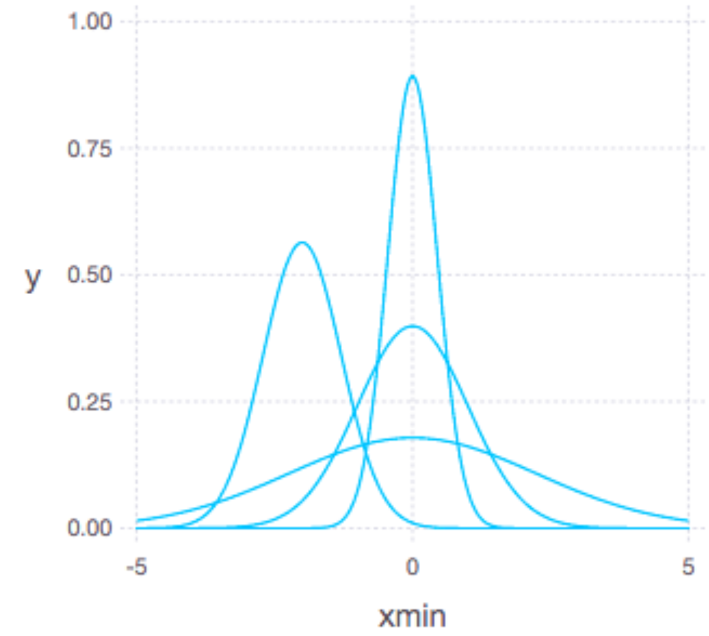
Dwell time - how long people spend on a web page

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

How to tell if dwell time differs from between versions

Normal (Gaussian) Distribution



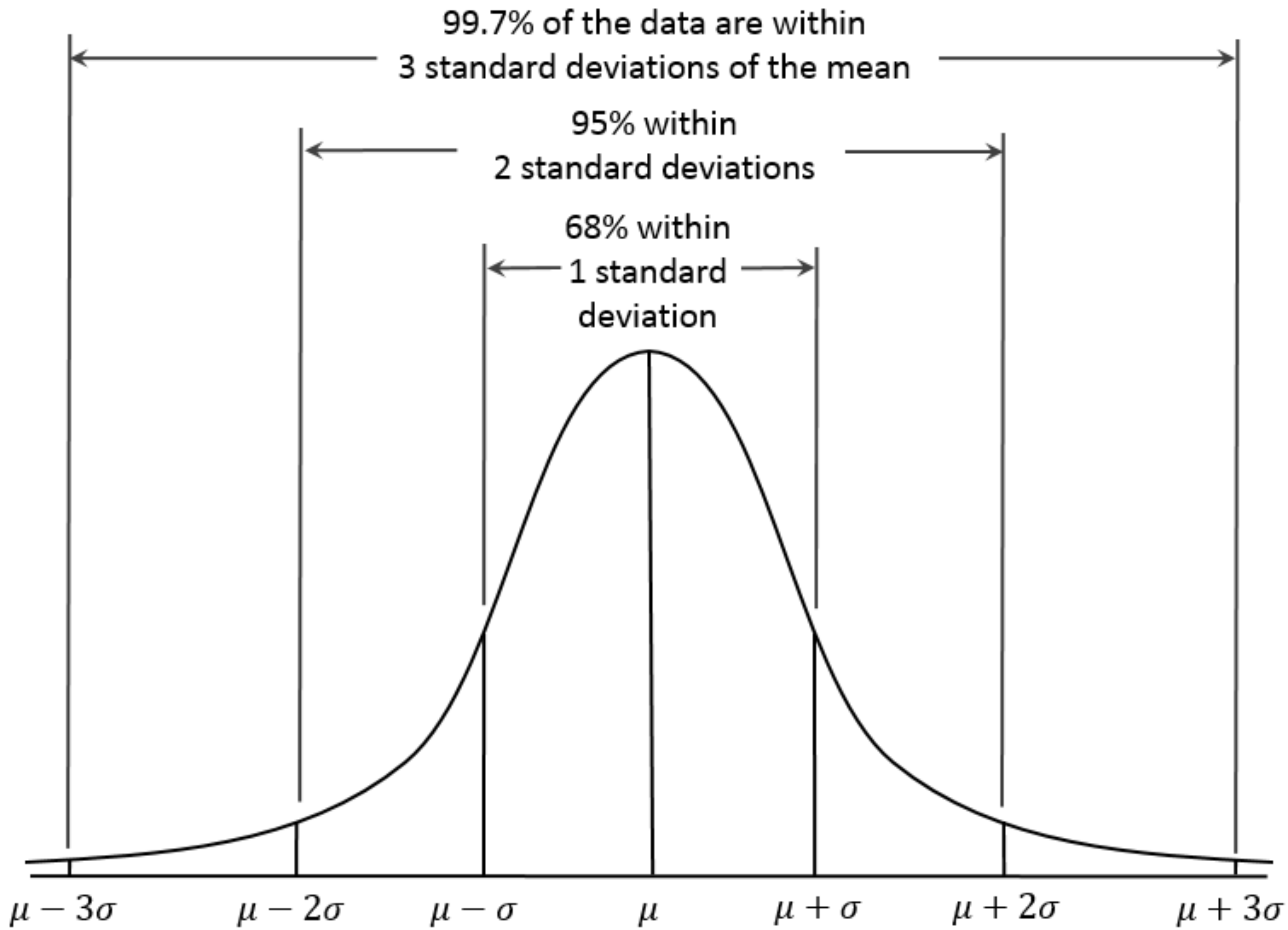


$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\sigma^2\pi}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Normal distribution is specified by

μ - mean, central point

σ - standard deviation



99.7% of the data are within
3 standard deviations of the mean
95% within
2 standard deviations
68% within
1 standard
deviation
$\mu - 3\sigma$
$\mu - 2\sigma$
$\mu - \sigma$
μ
$\mu + \sigma$
$\mu + 2\sigma$
$\mu + 3\sigma$

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 Error of sample = σ_x/sqrt(n)

Standard Error of mean (SEM)

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 occurring 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

Bonus Slide

Center For Open Science https://cos.io/our-services/research/

Reproduced 100 published Psychology studies

97 original studies had significant results $p < .05$

36 reproduced studies had significant results $p < .05$

47 original effects sizes were in the 95% confidence interval of replication

Bonus Slide - 2

John P. A. Ioannidis Why Most Published Research Findings Are False

PLoS Med. 2005 Aug; 2(8): e124.

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1182327/

Dr. Russell Schierling

Why Can't We Reproduce Biomedical Research?

https://bit.ly/2X15u03

Replication crisis

https://en.wikipedia.org/wiki/Replication_crisis

primarily affecting parts of the social and life sciences in which scholars have found that the results of many scientific studies are difficult or impossible to replicate or reproduce on subsequent investigation

Sensitivity & Specificity

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

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

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

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

Random sample

- 100 items
- Mean 1000 - 20
- Standard deviation 30

Compute the confidence interval for mean

Repeat 10 times

```
using Distributions
using HypothesisTests

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

10 Samples

Confidence interval of mean

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

Random sample

- 100 items
- Mean 1000 - 10
- Standard deviation 30

Compute the confidence interval for mean

Repeat 10 times

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

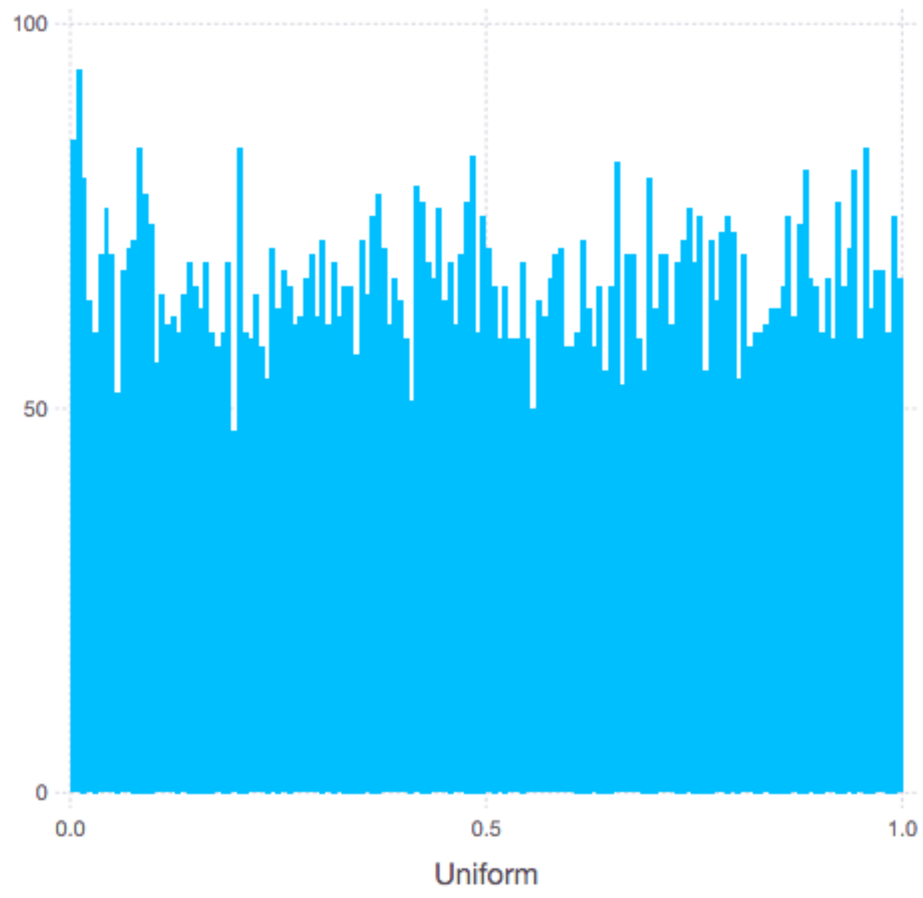
```
using Distributions
using HypothesisTests

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

Central Limit Theorem

Plot 10_000 random integers

Between 0 and 1

Bin the results



Central Limit Theorem

Let

$X_1, X_2, \ldots, X_N$ random sample

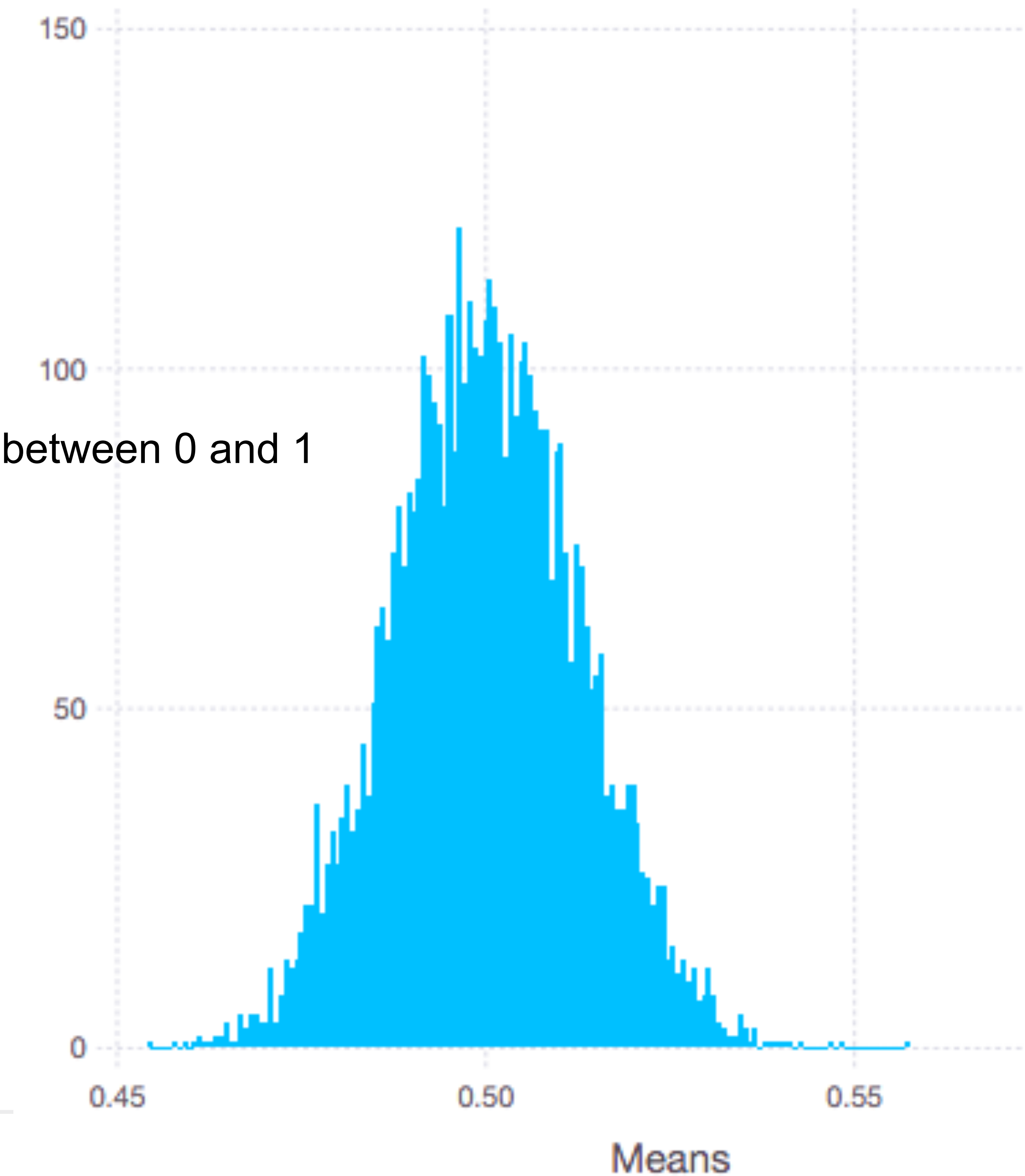
$S_N = (X_1 + \ldots + X_N)/N$

Then as N gets large S_N approximates the normal distribution

Compute the mean of 500 random numbers between 0 and 1

Repeat 5000 times

Plot the sums



Poincare’s Baker - Part Two

After being fined the baker still cheated

But always gave Poincare the heaviest loaf

Poincare still caught him!

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

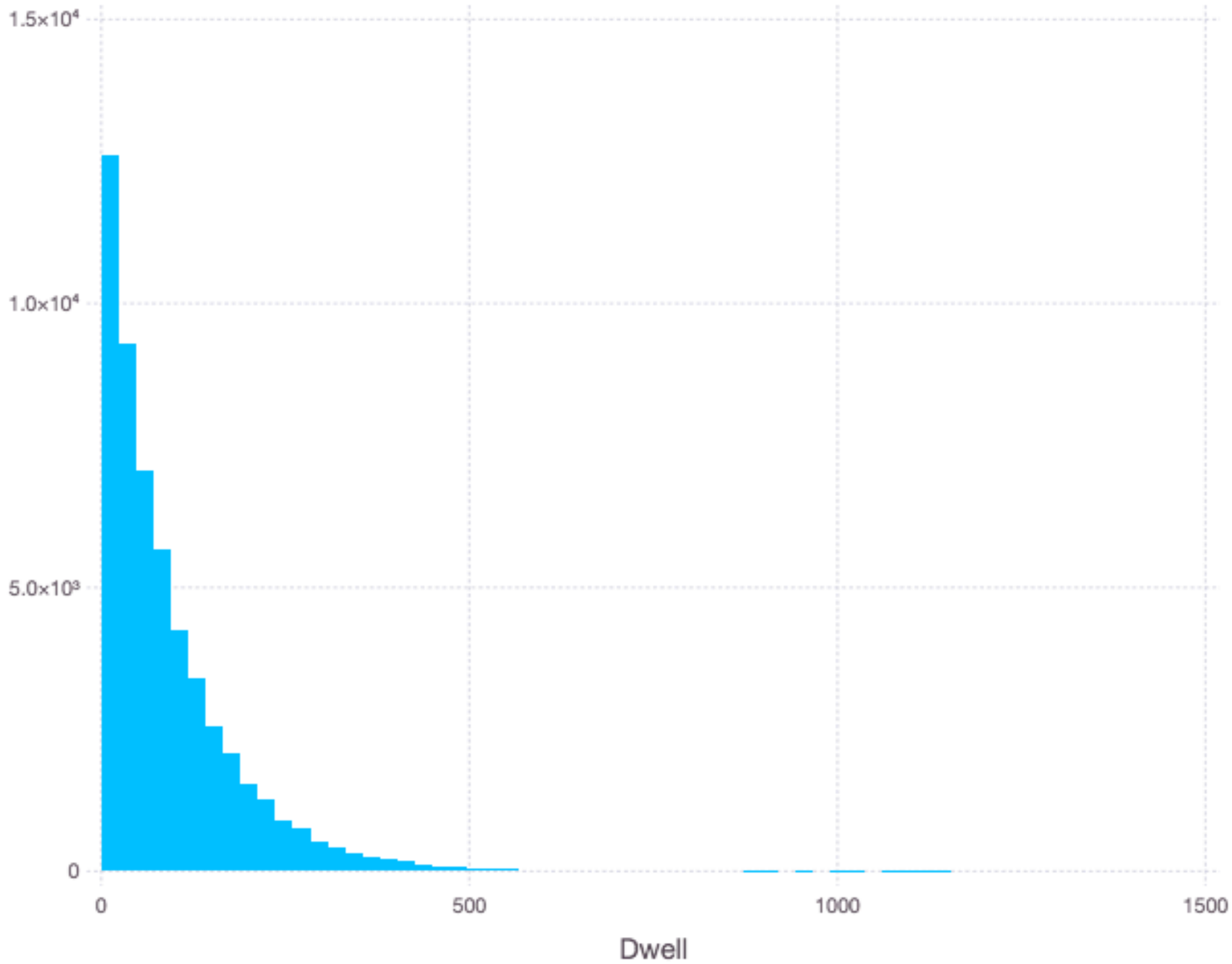
Dwell Data

```
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    |
```

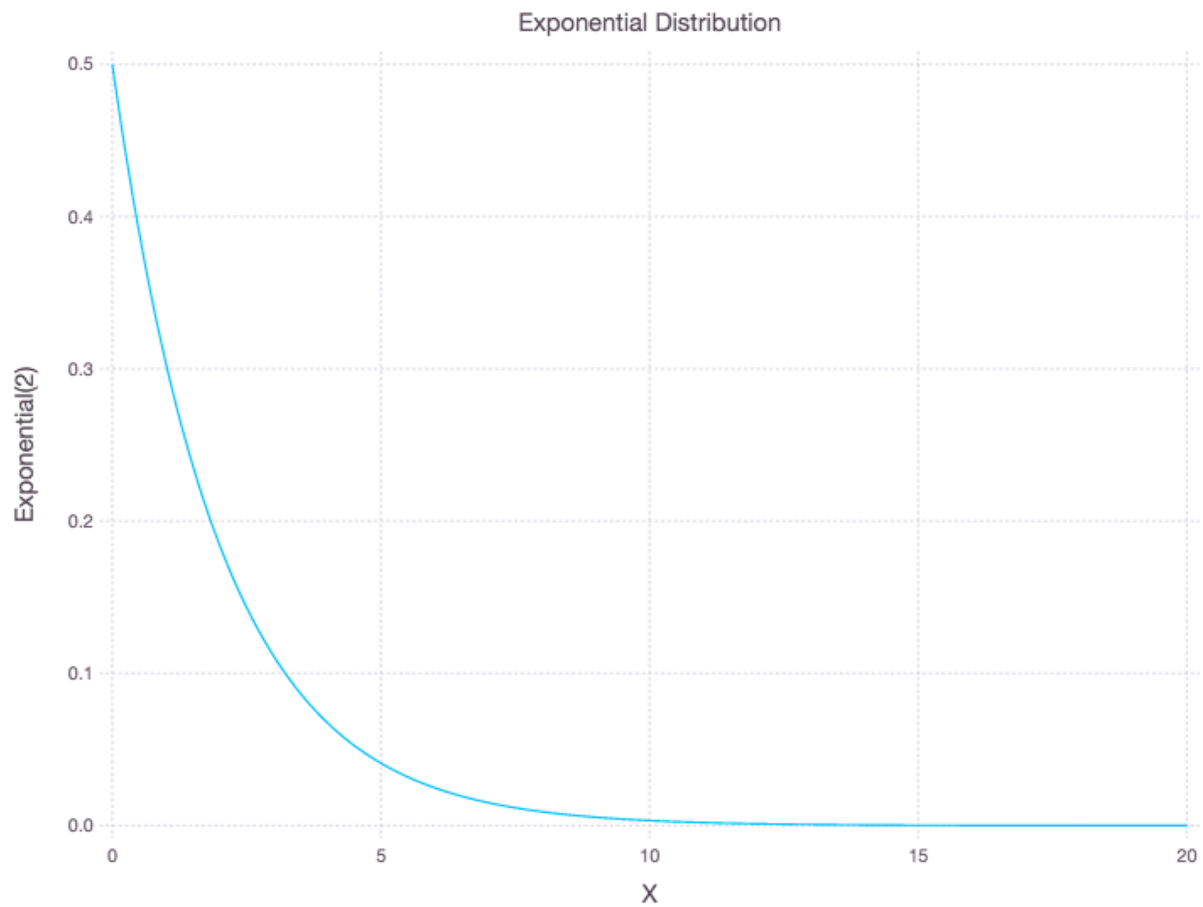
Dwell Times

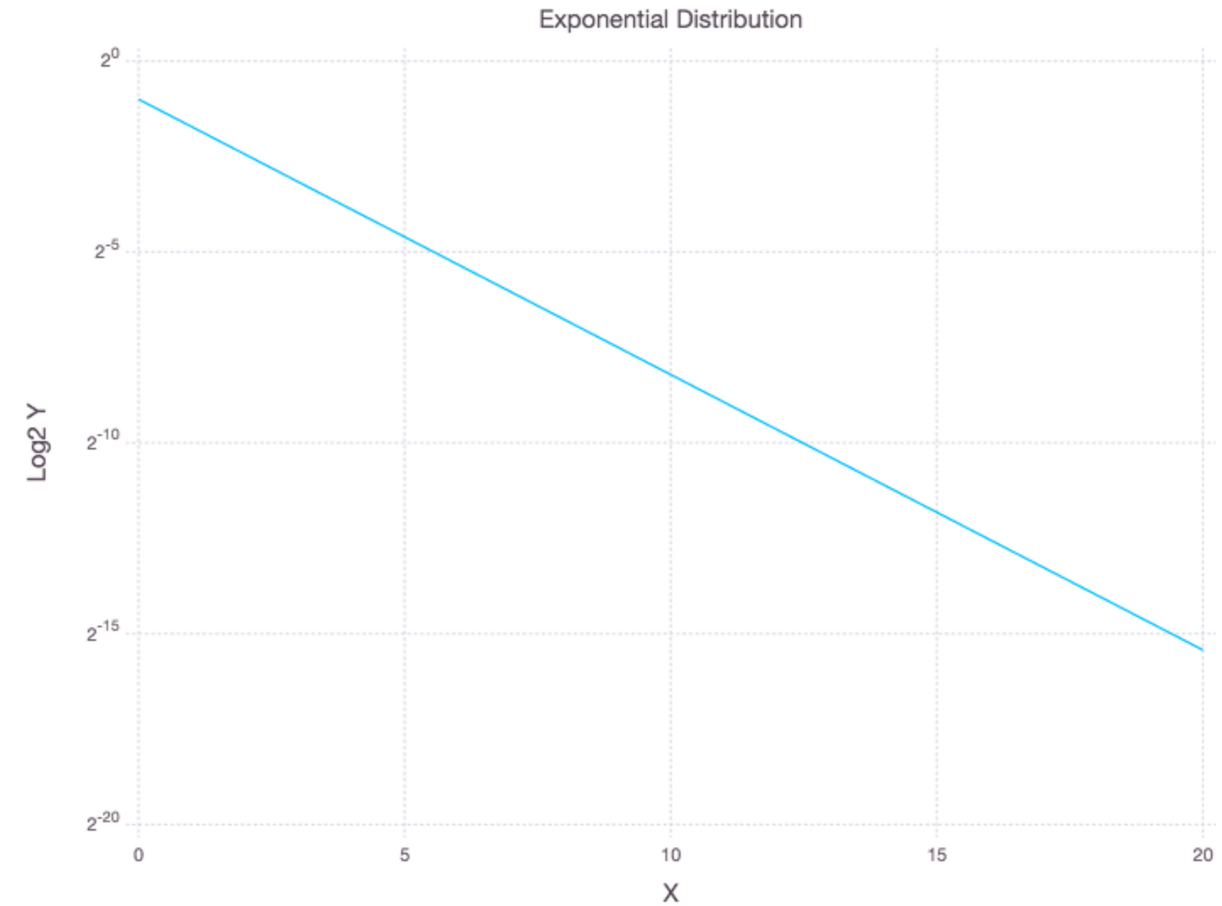
Divide the range of data into 50 equal bins

Plot the number of items in each bin



Exponential Distribution

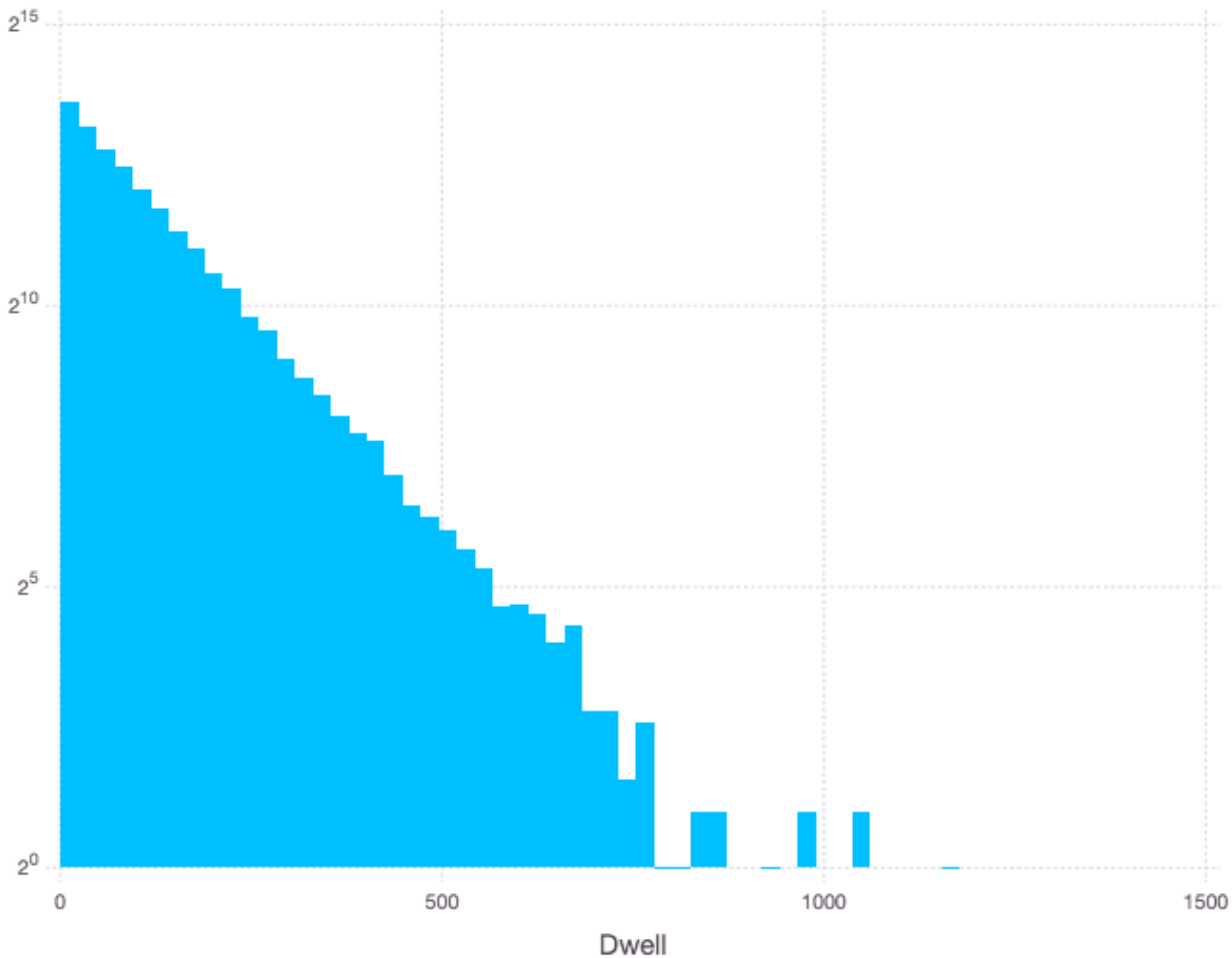




Log2(Y)

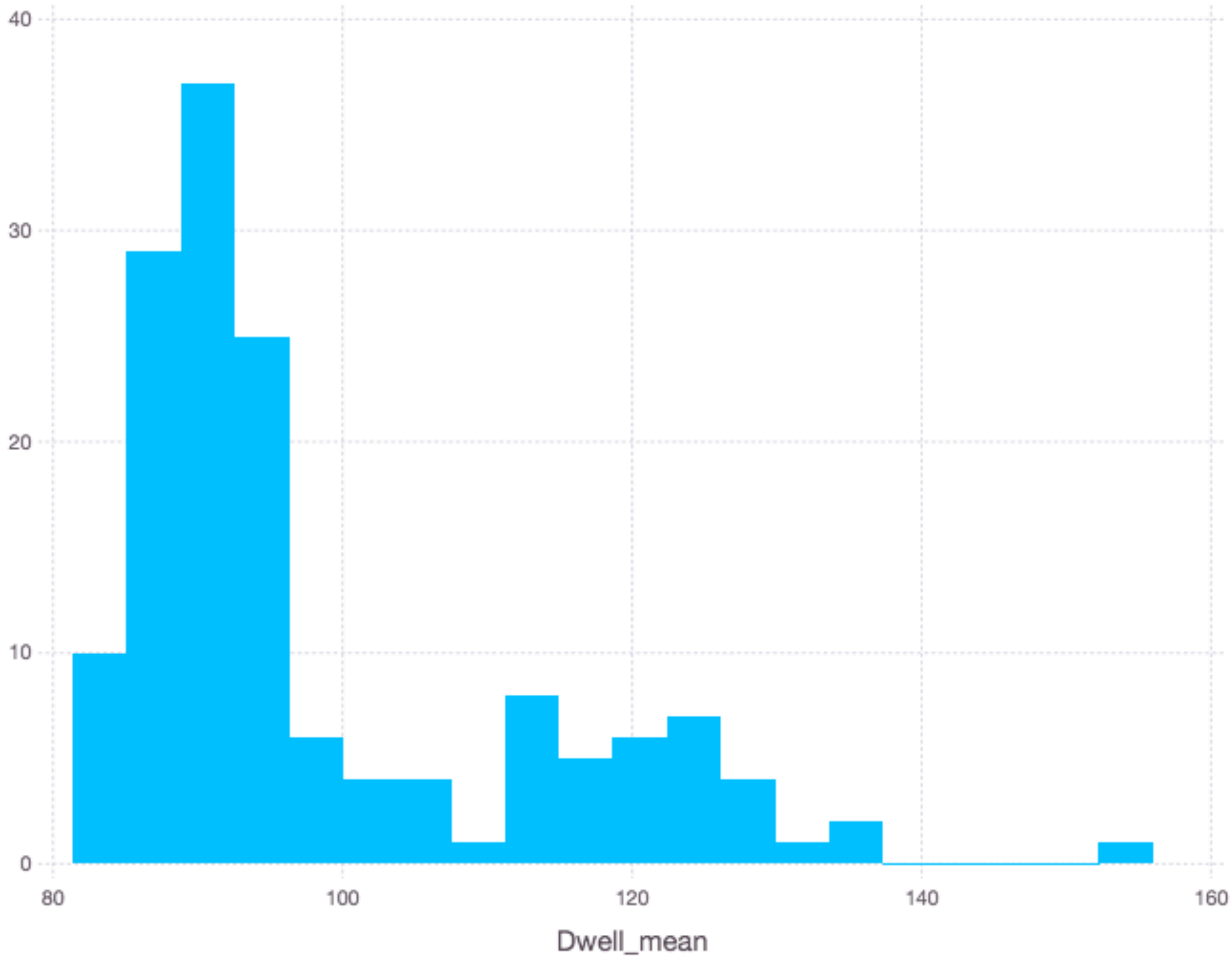
Log Scale - So Dwell Time is Exponential Dist.

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

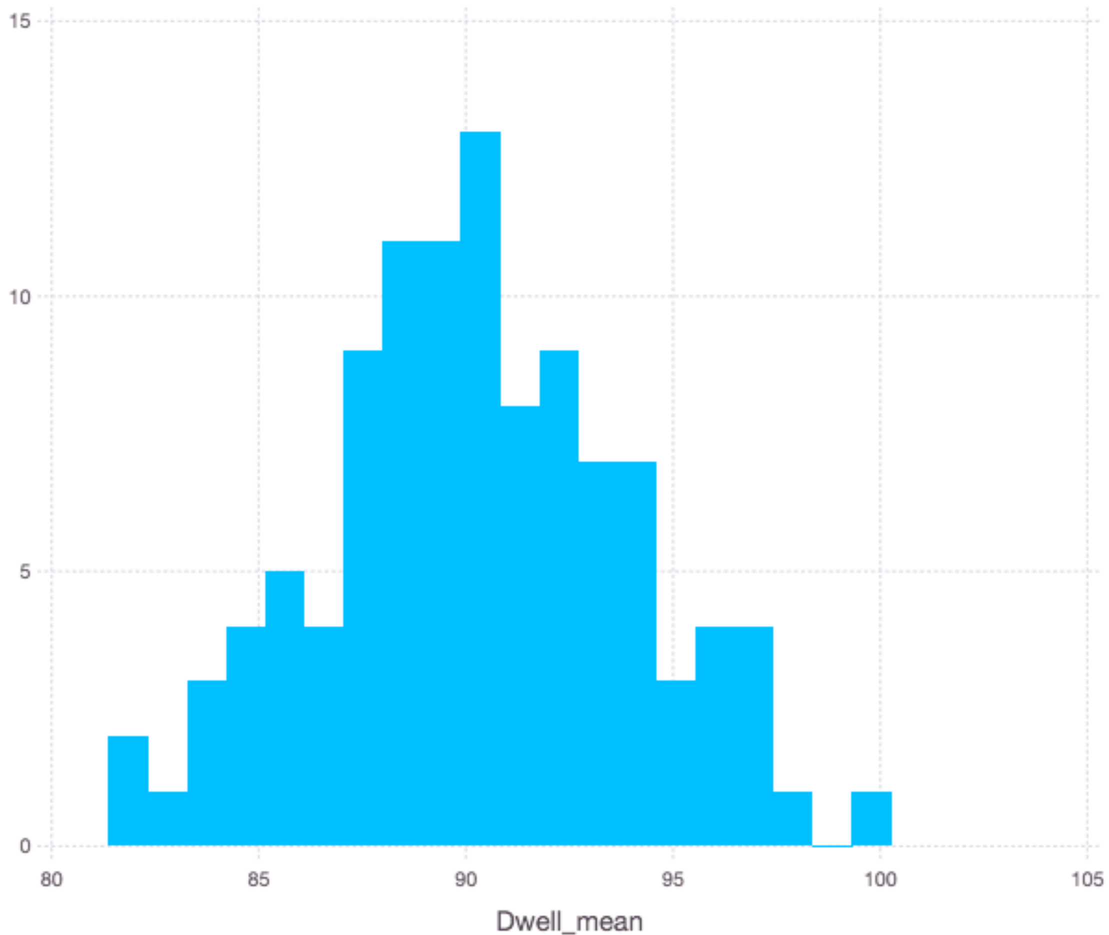


Compute Daily Mean Dwell Time

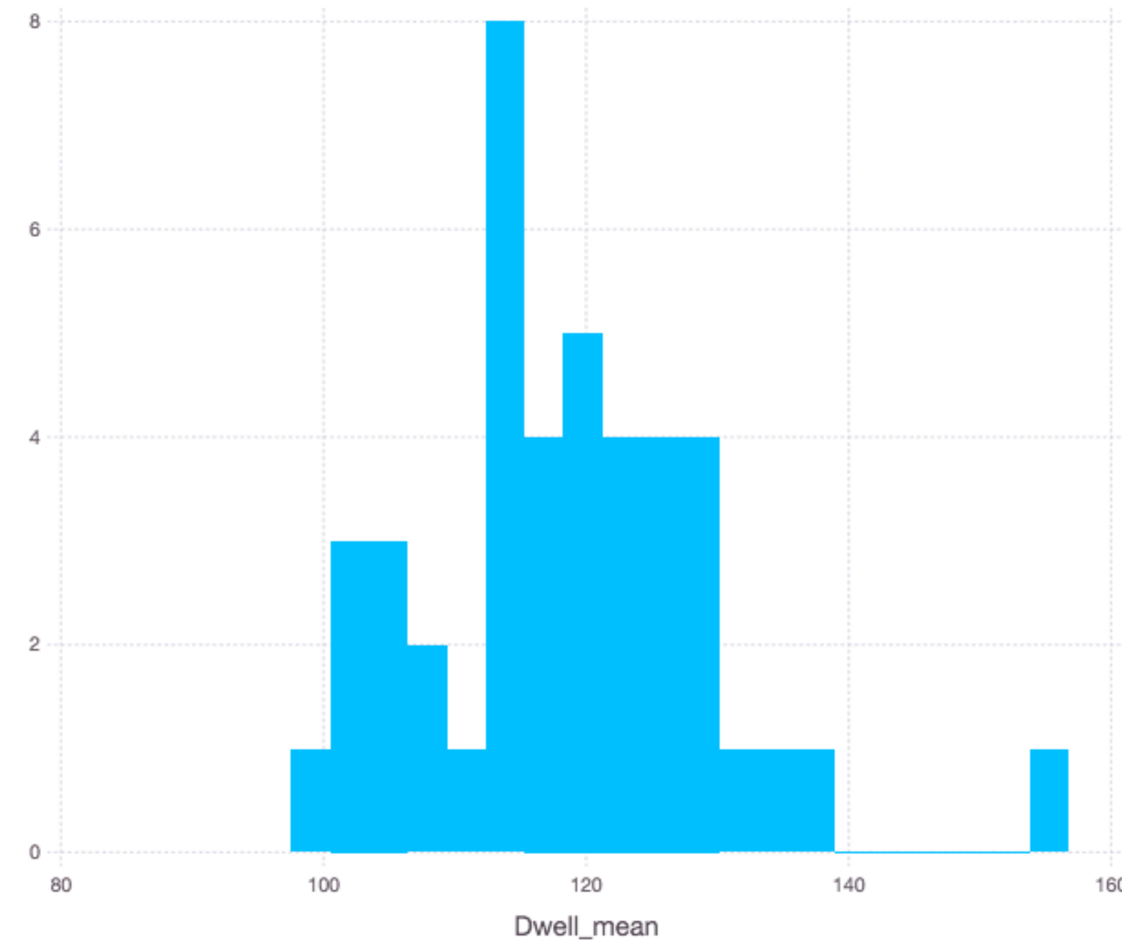
```
plot(daily_dwell, x="Dwell_mean", Geom.histogram(bincount=20))
```



Week Days



Weekends



sample size = 107

mean = 90.2

std = 3.7

CI of mean p = 0.05 (115,122)

sample size = 107

mean = 118.3

std = 11.0

CI of mean p = 0.05 (89.5 ,90.9)

Sampling - Motivation

How to find mean and median of 1 Billion values?

Web browser wants to warn user when they request a known malicious website

- Could be millions of malicious websites
- Don’t want to check server for each URL

Web Crawler

- Visit page A
- Extract all links from page A
- Repeat process on all links from page A
- How to know if you have already visited a page?
- Google indexes ~45 Billion web pages

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 Error of sample = σ_x/sqrt(n)

Standard Error of mean (SEM)

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

Sampling

100,000 data points

Compute the average

Take random sample of 1000 compute average

How close will sample average be to actual average?

Let s = average of the sample

n = sample size = 1000

Standard Error = standard deviation = s/sqrt(n)

Sampling

Let   s = average of the sample
n = sample size = 1000

Standard Error = standard deviation = s/sqrt(n)

Confidence Interval   (s - z*s/sqrt(n), s + z*s/sqrt(n))

Width of confidence interval = s + z*s/sqrt(n) - (s - z*s/sqrt(n))
= s + z*s/sqrt(n) - s + z*s/sqrt(n)
= z*s/sqrt(n) + z*s/sqrt(n)
= 2z*s/sqrt(n)

Sampling

Confidence Interval (s - z*s/sqrt(n), s + z*s/sqrt(n))

Experiment

100,000 random integer between 0 and 1000

Sample size 1,000

Sample mean (s) = 532.33

Confidence Interval at 95% = (499.3, 565.3)

Actual mean = 501.4

What if we want sample to be within 10?

Width of confidence interval = W = 2z*s/sqrt(n)

$n = 4z^{*2}s^2/W^2$
$= 4 * 1.96^2 * 501.4^2/10^2$
≈ 39000

Mean of samples of size 39000

502.37
500.795
503.108
502.488
499.351
499.907
500.791
501.248
501.814
501.707
⋮
504.143
500.595

Population mean

501.4

Bloom Filter

Burton Bloom - 1970

Space-efficient probabilistic data structure

Test whether an element is in a set

Bloom filter does not contain the elements in the set

False positive matches are possible

- Possibly in set

False negatives are not possible

- Definitely not in set

Types of Errors

False Positive (FP), type I error

Accepting a statement as true when it is not true

False Negative (FN), type II error

Accepting a statement as false when it is true

Bloom Filter - How it works

Empty Bloom filter

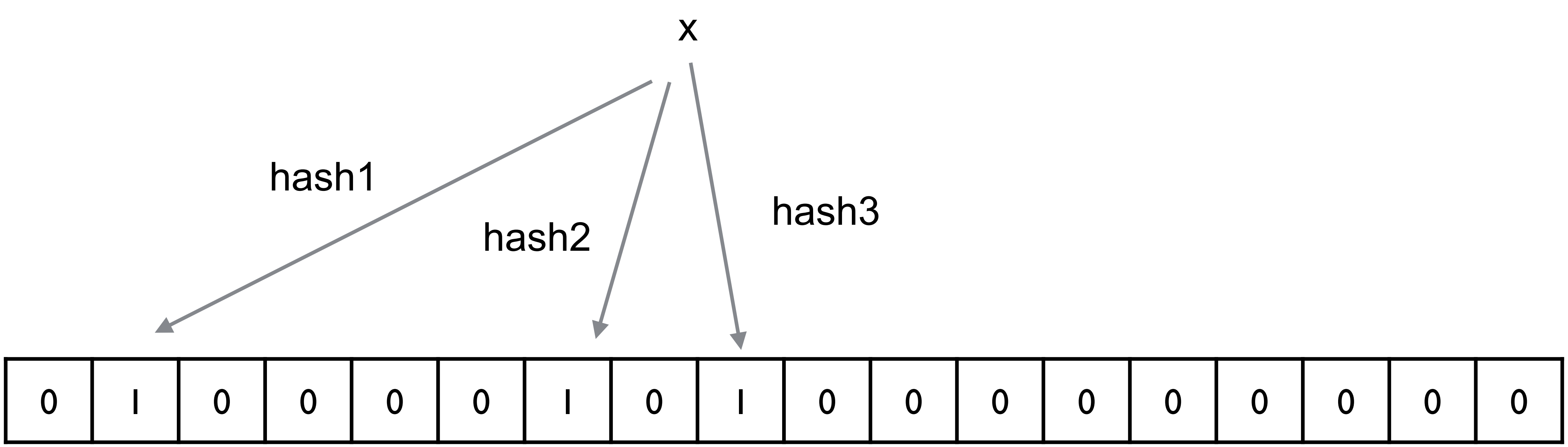
- m bits all 0
- k different hash functions

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bloom Filter - How it works

m = 18
k = 3

Insert x

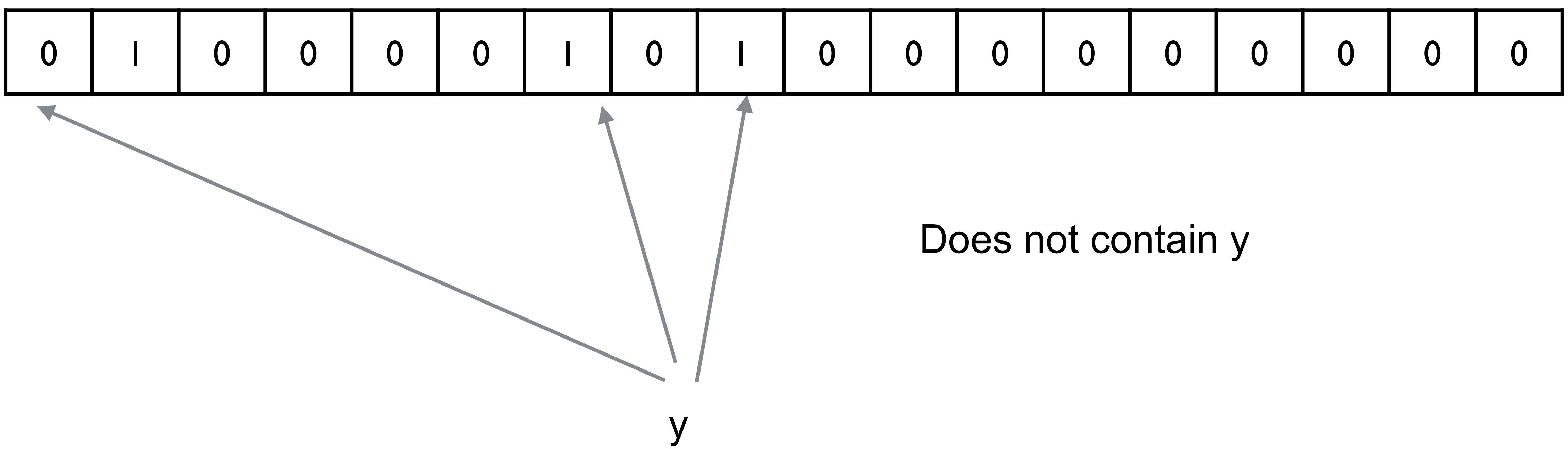


Bloom Filter - How it works

m = 18
k = 3

Contains y?

{x}



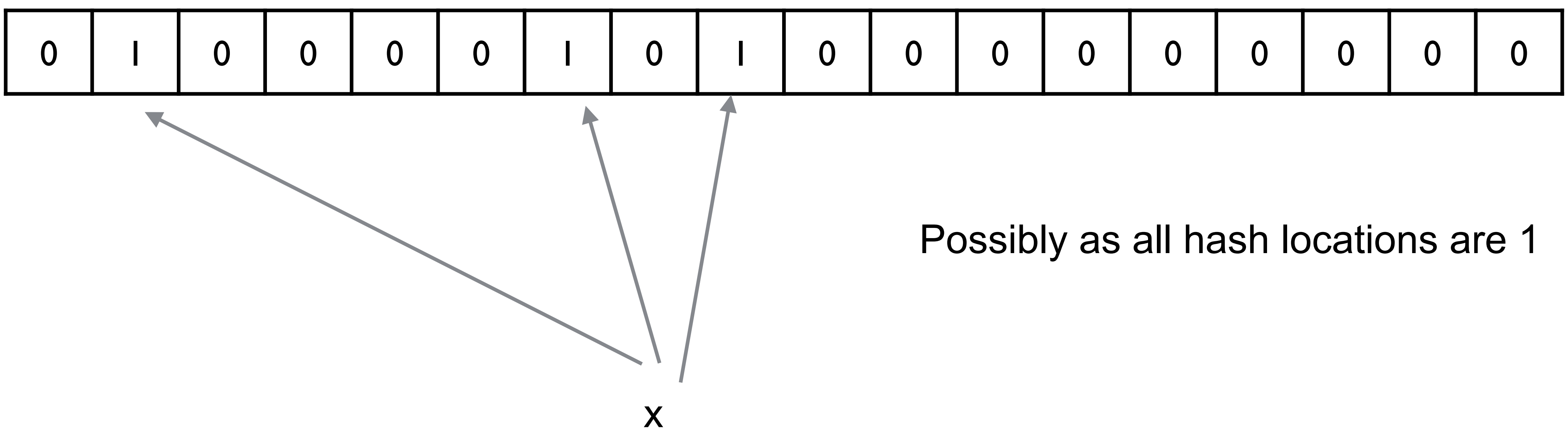
Does not contain y

Bloom Filter - How it works

m = 18
k = 3

Contains x?

{x}



Possibly as all hash locations are 1

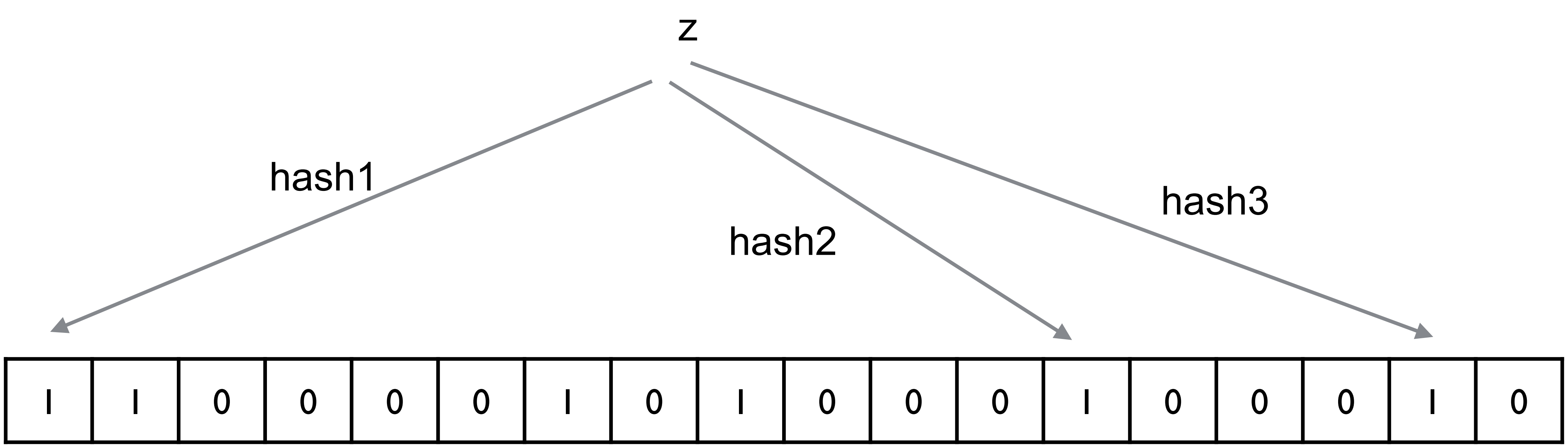
Bloom Filter - How it works

$m = 18$

$k = 3$

Insert z

{x}

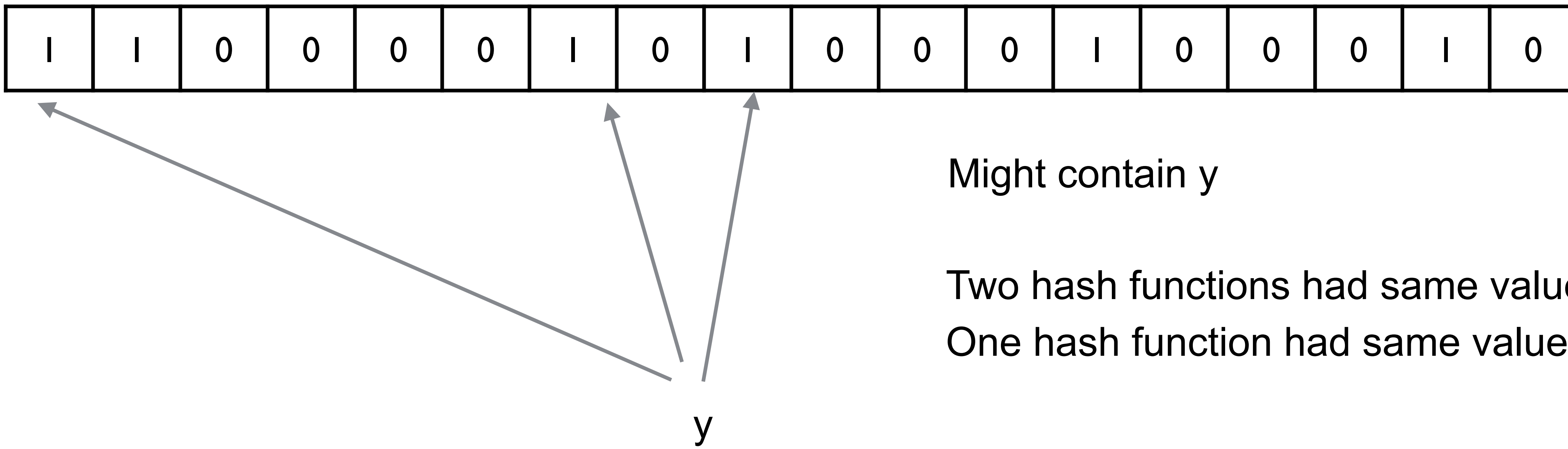


Bloom Filter - How it works

m = 18
k = 3

Contains y?

{x, z}



Might contain y

Two hash functions had same value as x
One hash function had same value of z

Bloom Filter - How it works

Larger m

- Decreases false positives
- Increases table size - fewer collisions

Larger k

- Decreases false positives up to a point
- But fills table faster

Bloom filter for Scala

https://github.com/alexandrnikitin/bloom-filter-scala

```
// Create a Bloom filter
val expectedElements = 1000000
val falsePositiveRate = 0.1
val bf = BloomFilter[String](expectedElements, falsePositiveRate)

// Put an element
bf.add(element)

// Check whether an element in a set
bf.mightContain(element)

// Dispose the instance
bf.dispose()
```

Bloom Filter - Sample Uses

Akamai's web servers

- Some pages are only accessed once - One-hit-wonders
- Only cache web page after second time it is accessed
- Use bloom filter to determine if page has been seen before

Google BigTable, Apache HBase and Apache Cassandra, and Postgresql

- Use Bloom filters to see if rows or columns exist
- Avoid costly disk access on nonexistent rows

Google Chrome web browser

- Use Bloom filter to identify malicious URLs
- If filter contains the url then check server to make sure

Medium

- Uses Bloom filters to avoid recommending articles a user has previously read

Heavy Hitters Problem

Streaming

Real time

Computing popular products

- Given the page views on Amazon which products are viewed the most?

Computing frequent search queries

- Given the stream of Google searches what are the popular searches
- 3.5 billion searches per day

View Tweets

- How often are trees viewed? What the most popular tweets?

Heavy Network flows

- Given packet count source and destination through switch
- Where is the traffic the heaviest?
- Cisco Nexus 9500 - 172.8 Tbps
- Useful to detect DoS attacks

Volatile Stocks

- Given stream of stock transactions which stocks are
  - Traded the most
  - Change prices the most

Count-Min Sketch

Graham Cormode and S. Muthu Muthukrishnan - 2003

Consume a stream of events

Count the frequency of the different types of events in the stream

Does not store the events

Counts for each event type

- Estimate of actual count
- Within given range of actual count with given probability

Count-Min Sketch - How it works

Initial count-min sketch

- w - columns
- d - rows
- d different hash functions
- All entries integers = 0

w determines

- Interval length containing actual count

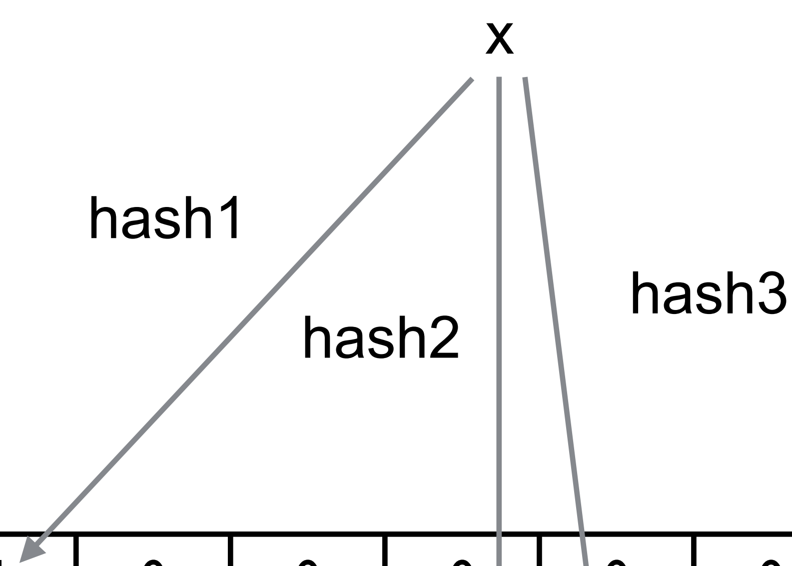
d determines

- Probability that actual count is in interval

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Count-Min Sketch - How it works

Event x



0	1	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0

Count-Min Sketch - How it works

Event y

y

0	1	1	0	0	0	0	0	0	0
0	0	0	0	2	0	0	0	0	0
0	0	0	0	0	1	0	0	1	0

Count-Min Sketch - How it works

Event x

x

0	2	1	0	0	0	0	0	0	0
0	0	0	0	3	0	0	0	0	0
0	0	0	0	0	2	0	0	1	0

Count-Min Sketch - How it works

Event z

z

0	2	2	0	0	0	0	0	0	0
0	0	0	1	3	0	0	0	0	0
0	1	0	0	0	2	0	0	1	0

Count-Min Sketch - How it works

How often did x occur?

Look at counts for x in each row

Return the minimum count

x

0	2	2	0	0	0	0	0	0	0
0	0	0	1	3	0	0	0	0	0
0	1	0	0	0	2	0	0	1	0