Final Review

Max Turgeon

STAT 3150–Statistical Computing

Main theme

- Recall the main theme of the course: **using computational techniques to solve statistical problems**.
- What kind of statistical problems?
    - Point estimation
    - Interval estimation
    - Hypothesis testing

Numerical methods and Optimisation

- For the first two modules, we specifically looked at point estimation.
- We talked about the following methods:
    - Bisection/Brent's method for root finding in one dimension.
    - Newton-Raphson for optimisation in any dimension.

Generating random variates

- R has many built-in functions for generating random variates.
    - `runif`, `rnorm`, etc.
- We discussed general techniques when these functions aren't enough.
    - Inverse transform, or generally any type of transformation.
    - Accept-reject sampling.
- When would you need to generate random variates?
    - Estimate expected values (i.e. Monte Carlo integration)
    - Estimate probability statements
    - Simulation studies

Accept-Reject algorithm i

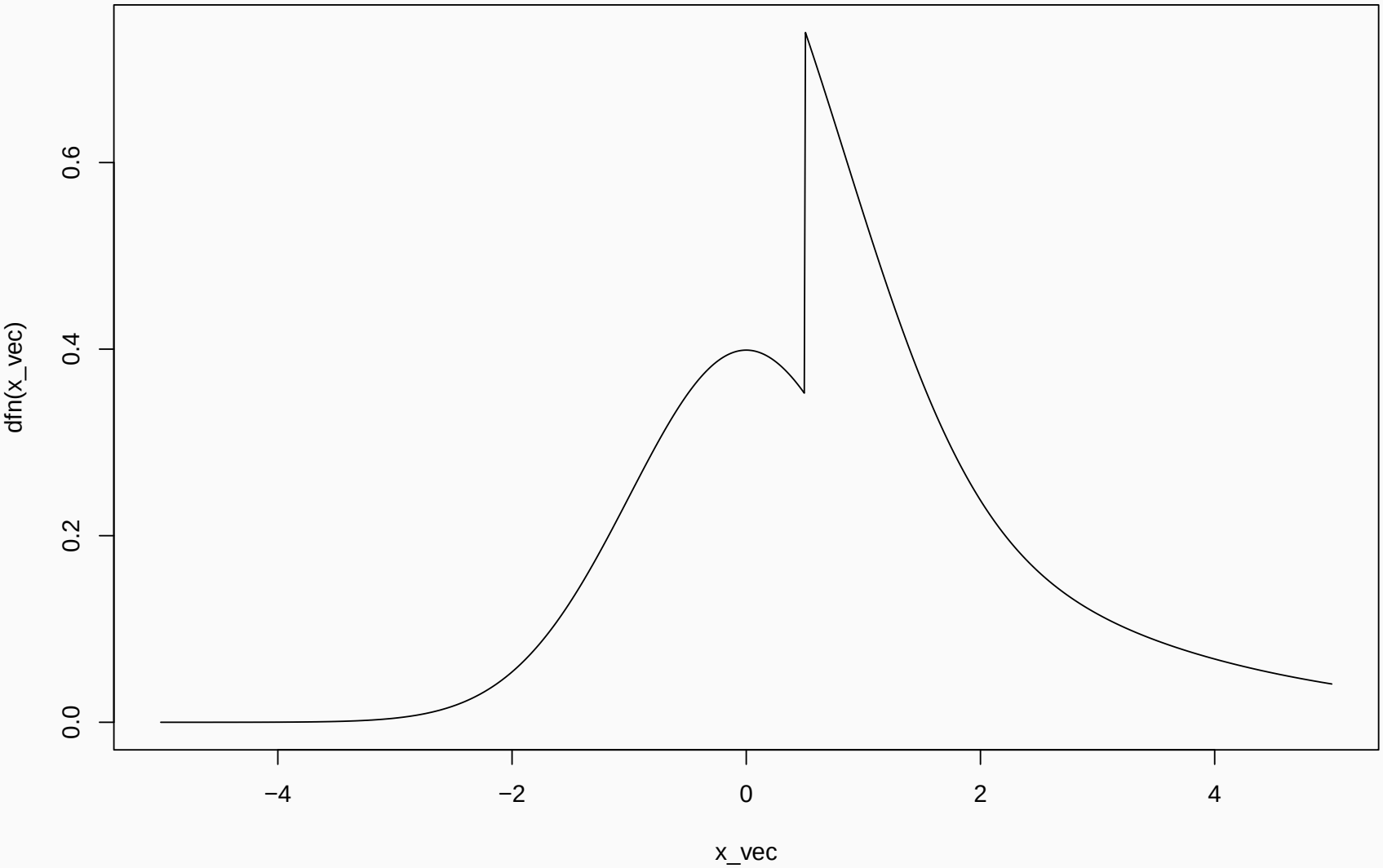
- Recall the general goal: we want to sample from a distribution with density f.
- Instead, we sample from a "dominating" density g, and then randomly decide if we keep the sample or not.
  - **Dominating**: There exists a constant $c > 1$ such that $\frac{f(t)}{g(t)} \leq c$ for all t such that $f(t) > 0$.
  - In particular, the support of g must include that of f (it's best if they are equal).
- To decide if we keep the sample, we sample u from a uniform $U(0, 1)$ and accept if $u < \frac{f(y)}{cg(y)}$.
- Suppose we want to sample from a complicated distribution:

Accept-Reject algorithm ii

```r
dfn <- function(x){
    out <- numeric(length(x))
    out[x <= 0.5] <- dnorm(x[x <= 0.5])
    out[x > 0.5] <-  dnorm(x[x > 0.5]) +
        dchisq(x[x > 0.5], df = 2)
    out
}

# Let's visualize
x_vec <- seq(-5, 5, length.out = 1000)
plot(x_vec, dfn(x_vec), type = 'l')
```

Accept-Reject algorithm iii

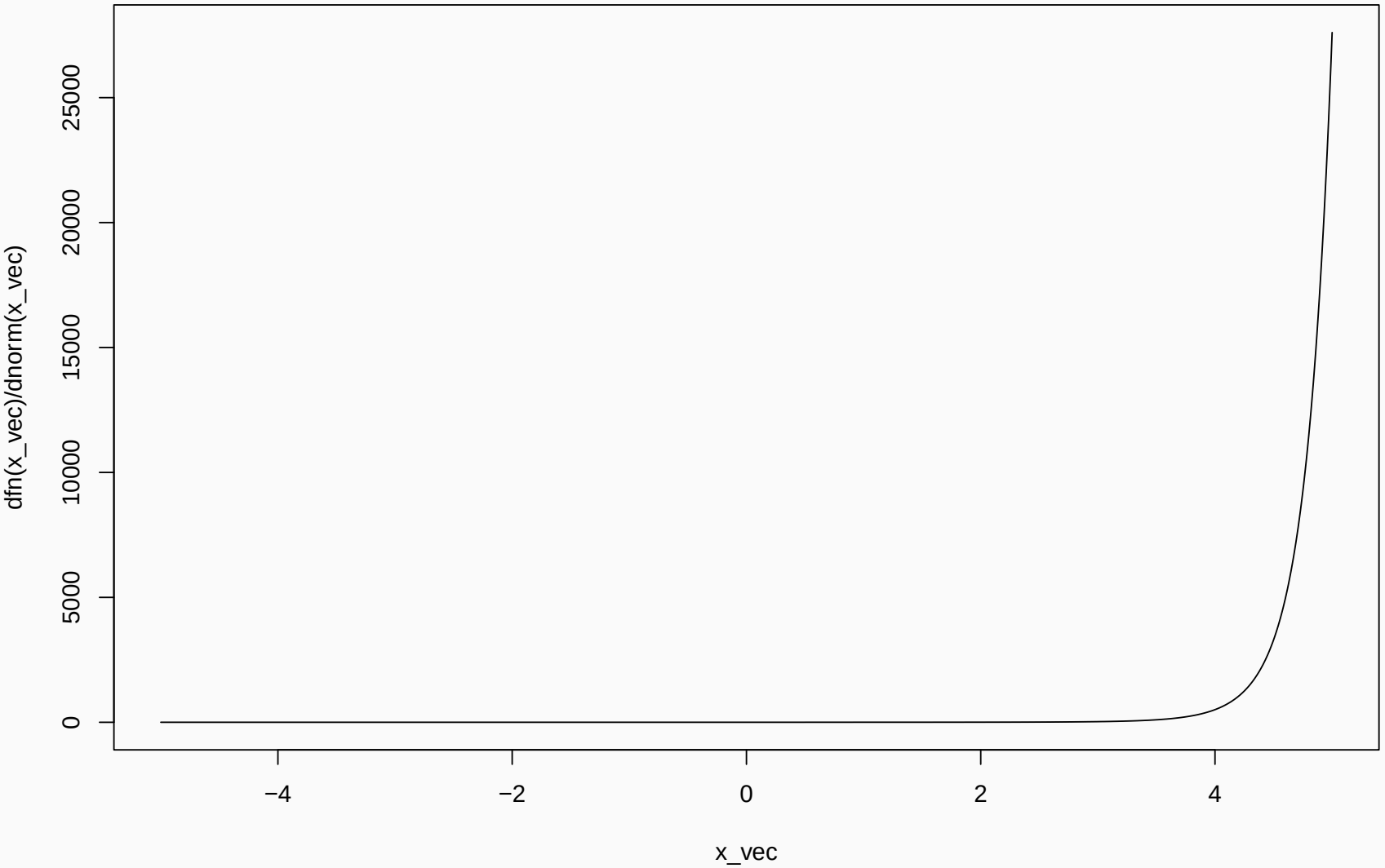


Accept-Reject algorithm iv

- The support is the whole real line. This gives us some natural candidates for the dominating distribution:
  - Normal distribution
  - t distribution
- Goal: Find a dominating (or proposal) distribution with "heavier tails" so that the ratio $f(t)/g(t)$ is bounded.
- Let's try the normal density.

```
# Visualize the ratio
plot(x_vec, dfn(x_vec)/dnorm(x_vec), type = 'l')
```

Accept-Reject algorithm v

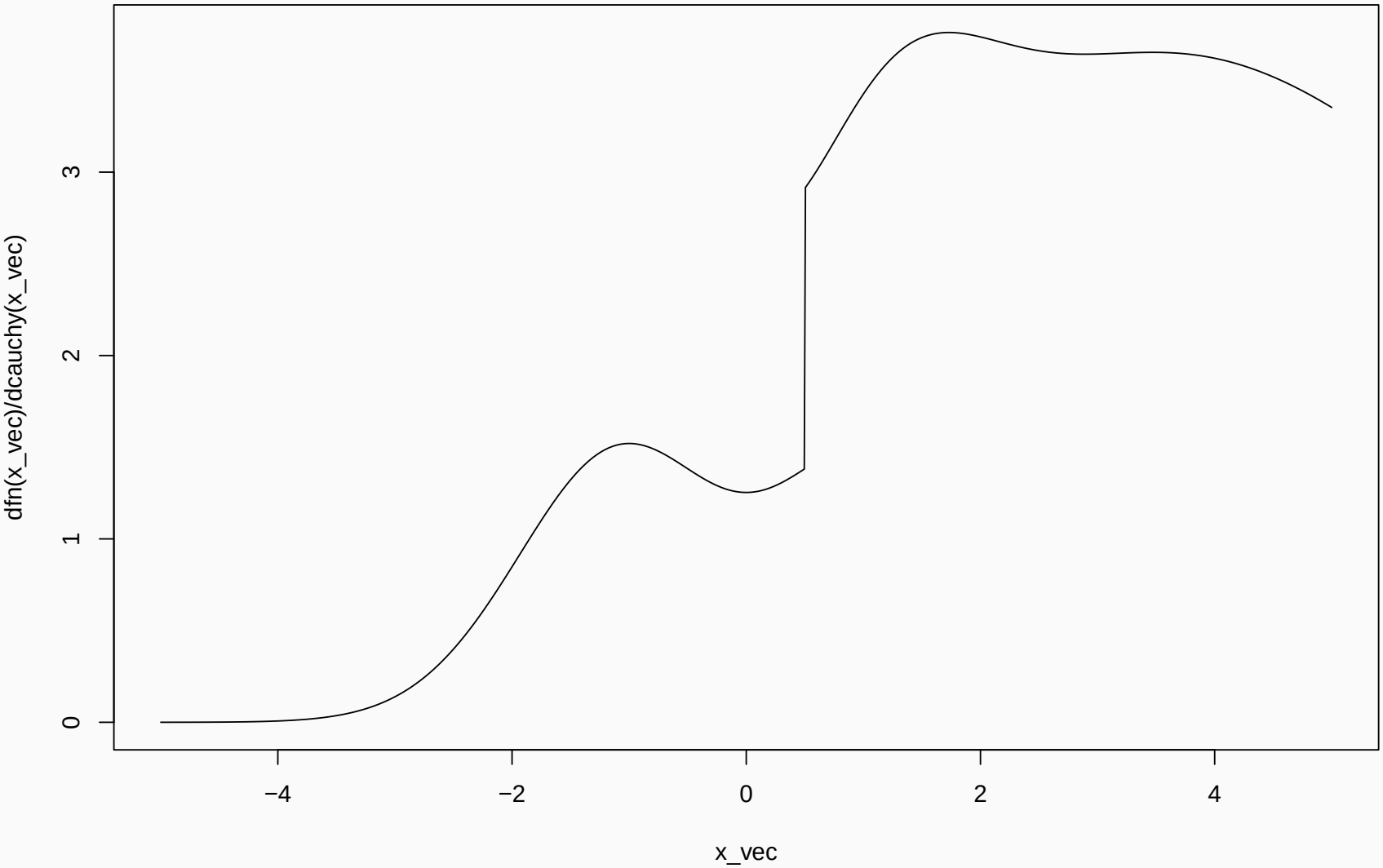


Accept-Reject algorithm vi

- The normal density will not work, because the ratio is unbounded...
- Let's try the Cauchy distribution (aka $t(1)$):

```
# Visualize the ratio
plot(x_vec, dfn(x_vec)/dcauchy(x_vec), type = 'l')
```

Accept-Reject algorithm vii



Accept-Reject algorithm viii

- It looks like the ratio is bounded above by $c = 4$!
- We can double-check using calculus: if the ratio is bounded, then it will attain a maximum, which corresponds to when the derivative is equal to 0.
- The density of interest is piece-wise defined: we would technically need to optimize on both pieces, but from our figure we know the maximum will be on the second piece (e.g. $x > 0.5$).
    - We will omit the details. You can either solve with Calculus, or use a numerical method to find a more exact c.

Accept-Reject algorithm ix

```r
# Set parameters----
C <- 4 # Constant
n <- 1000 # Number of variates
k <- 0 # counter for accepted
j <- 0 # iterations
y <- numeric(n) # Allocate memory
```

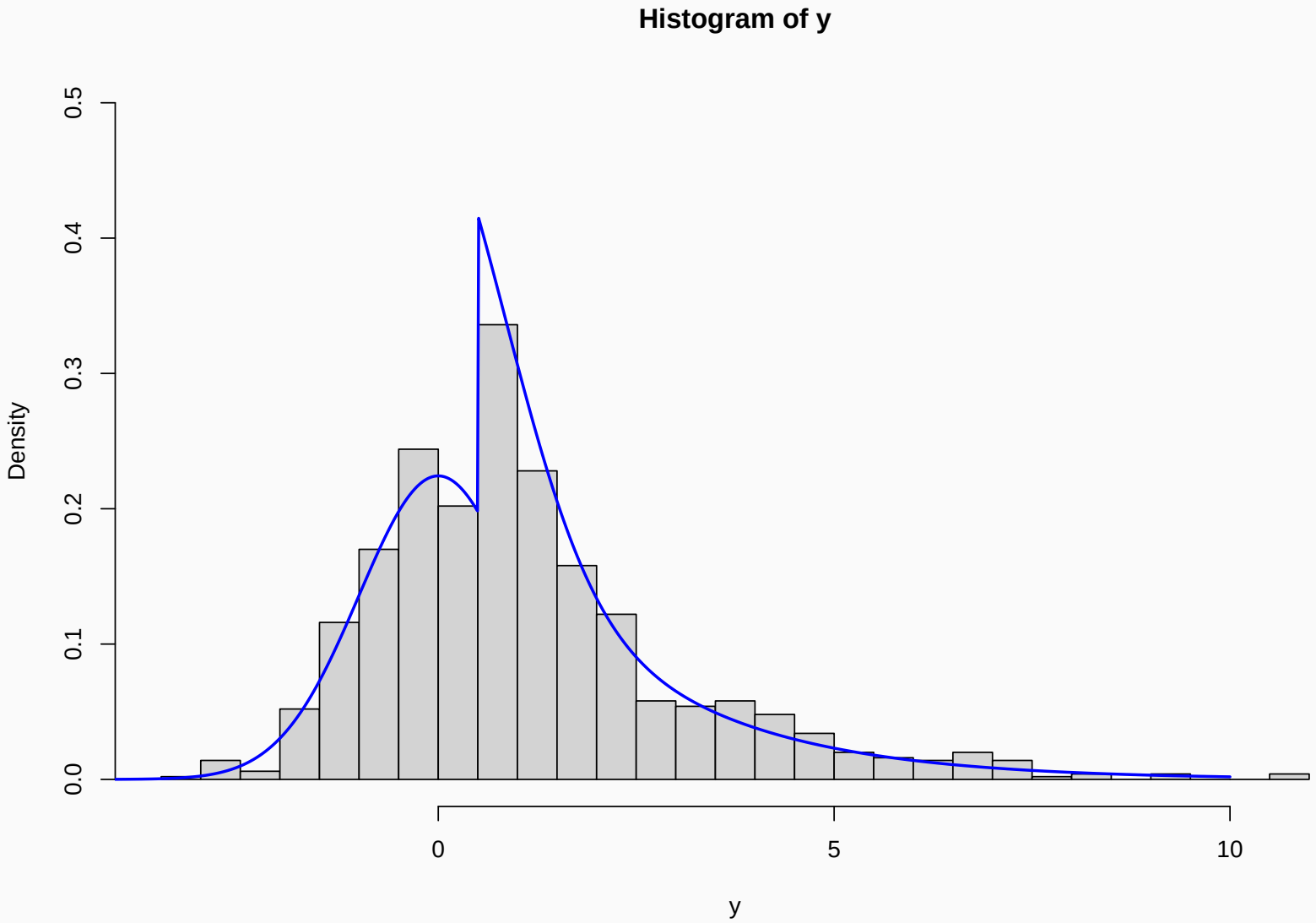
Accept-Reject algorithm x

```
while (k < n) {
  u <- runif(1)
  j <- j + 1
  x <- rcauchy(1) # random variate from g
  if (u < dfn(x)/(C*dcauchy(x))) {
    k <- k + 1
    y[k] <- x
    }
}
```

Accept-Reject algorithm xi

```
# Visualize the histogram
hist(y, 50, freq = FALSE)
```

Accept-Reject algorithm xii



Accept-Reject algorithm xiii

- **Note**: We didn't even start with an actual density; it doesn't integrate to 1.
- But the algorithm still works (proof left as an exercise).

Monte Carlo integration i

- This topic mostly falls under *point estimation*.
- Estimate quantities of the form

$$E(g(X)) = \int g(x)f(x)dx, \qquad X \sim f.$$

- Trace plot = diagnose convergence issues
- Variance reduction
  - Antithetic variables
  - Control variates
  - Importance sampling
- Confidence intervals in MC integration are based on the Central Limit Theorem

Monte Carlo integration ii

- Since our estimates are sample means, we need to divide by $\sqrt{n}$, where n is the number of variates in the sample mean.

- **When would you use MC integration?**
  - To estimate difficult integrals.
  - Many, many estimators can be defined as expected values of transformations $g(X)$ of a random variable X.

Importance sampling

- It's a form of **variance reduction** for Monte Carlo integration.
- Based on the following identity:

$$E_f(g(X)) = E_\phi\left(\frac{g(X)f(X)}{\phi(X)}\right),$$

  as long as ϕ is nonzero on the support of f.
- We want to choose the importance function ϕ such that:
  - ϕ is a density from which it is "easy" to sample.
  - the ratio $\frac{|g(X)|f(X)}{\phi(X)}$ is almost constant.
- **Why do we care so much about reducing variance anyway?**
  - Because smaller variance means smaller confidence intervals, which means more accurate inference.

How to integrate?

Where does MC integration fit within the different ways of doing integration?

Choose the first thing that works in this order:

1. Exact integration (works well for simple integrand)
2. Symbolic mathematics (e.g. Maple or Mathematica)
3. Numerical integration (works best in 1 dimension)
4. **MC Integration**
5. Markov Chain Monte Carlo (cf. Bayesian statistics)

Importance Sampling—Example i

- Consider the following integral:

$$\int_0^\pi \frac{dx}{x^2 + \cos^2 x}.$$

- How can we write this as an expected value $E_f(g(X))$?

Importance Sampling—Example ii

- We can take:
  - $g(x) = 1/(x^2 + \cos^2 x)$
  - $X \sim \text{Unif}(0, \pi)$

```
g_fun <- function(x) pi/(x^2 + cos(x)^2)
f_vars <- runif(1000, min = 0, max = pi)
mean(g_fun(f_vars))
```

```
## [1] 1.542862
```

Importance Sampling—Example iii

- In importance sampling, we now need an importance function $\phi(x)$, which is the density of a distribution we can sample from.
- Let's try ϕ the density of $Exp(1)$.
- **Important observation**: The density f of $\mathrm{Unif}(0, \pi)$ is zero outside the integral bounds, so our ratio $g(x)f(x)/\phi(x)$ will be zero as expected. If it wasn't, we would need to modify our implementation of g to make sure it is.

Importance Sampling—Example iv

```
phi_vars <- rexp(1000)
# Think of re-weighting
weights <- dunif(phi_vars,
                 max = pi)/dexp(phi_vars)
mean(g_fun(phi_vars)*weights)
```

```
## [1] 1.584207
```

Importance Sampling—Example v

- You could improve the performance of this estimator by sampling from a *truncated* Exponential.
- You could use an Exponential with a different mean (perhaps also truncated).
- These two are left as an exercise.

Monte Carlo methods for Inference

- This module was an interlude, connecting Monte Carlo integration and resampling methods.
    - What is a statistic? An estimator? A sampling distribution?
    - What is a type I error? Type II error? Power?
- If we are willing to completely specify the data generating mechanism, we can study the consequences of these assumptions through **Monte Carlo simulation**.
    - Which estimator is more efficient (i.e. has smallest variance)?
    - Does my confidence interval have the right coverage probability?
    - Which hypothesis test has largest power?

Resampling methods

- The next few modules were on resampling methods:
  - *Jackknife*: "Resample" all subsets of size $n - 1$.
  - *Bootstrap*: Resample n observations **with replacement**.
  - *Permutation tests*: Permute all observations to mimic resampling under the null hypothesis
- It's during these modules that we finally started analysing data.

Jackknife

- Mainly presented for its historical importance.
  - It can be formalized as a “linear approximation” to the bootstrap.
  - It also helps motivate some quantities/techniques, e.g. student residuals, Cook’s distance, leave-one-out cross-validation.
- We used it for estimating the **standard error** and **bias** of an estimate.
  - Use the formulas provided.
  - But it doesn’t always work! E.g. median, quantiles.
- We can construct confidence intervals using the CLT.
  - No general accuracy guarantees.

Bootstrap i

- Bootstrap is almost always preferable to jackknife.
  - It is valid under more general assumptions.
  - Can be used to construct valid confidence intervals.
- Recall the general idea: we are trying to mimic going back and collecting more samples.
  - **What should we bootstrap?** The quantities we sampled in the first place.
  - In practice, we need to resample rows of `data.frame`s, so that we *preserve* the correlation structure between the different measurements.
- We discussed 5 types of confidence intervals for bootstrap.

Bootstrap ii

- **Remember**: no need to divide by $\sqrt{n}$ with bootstrap.
- **Very important**: Throughout, we assumed simple random sampling.
    - In particular, we assumed the observations were independent.
    - Can be generalized with care.
- We spent considerable time talking about bootstrap and linear regression.
    - *Resample residuals*: Need to assume linearity, additivity, equal variance AND independence of the errors.
    - *Resample cases/rows*: Only need to assume independence.

Bootstrap iii

- If the only thing you remember from STAT 3150 after the final exam is how to *properly* use bootstrap, I'll be happy.
    - But it won't be enough to pass the final exam!

Permutation tests

- Jackknife and bootstrap are mainly used for point/interval estimation.
    - Of course, any confidence interval leads to a hypothesis test.
- Permutation tests are specifically for hypothesis testing.
- They are a family of **non-parametric** methods for testing

$$H_0 : F = G.$$

- In other words: two-sample tests of equal distribution.
- **Main idea**: if the data all come from the same distribution, then which ones are Xs and which ones are Ys is irrelevant.

Final words

- Statistical computing is deeply connected to modern statistics.
    - You **cannot** do statistics in the 21st century without computing.
- Some areas of statistics are more computational than others:
    - Statistical learning
    - Bayesian statistics
    - High-dimensional data