# Stat 343: Optimization in Stan

## Wind Energy

This example is adapted from Chihara and Hesterberg (2011). Here's a quote from them:

```
"[U]nderstanding the characteristics of wind speed is important.  Engineers use wind speed information t
```

```
The Weibull distribution is the most commonly used probability distribution used to model wind speed ..
```

If $X \sim \text{Weibull}(k, \lambda)$, then it has pdf

$$f(x|k, \lambda) = \frac{kx^{k-1}}{\lambda^k} e^{-(x/\lambda)^k}$$

In R, the density function can be evaluated with the `dweibull` function, which has the following arguments:

- `x`: vector of quantiles.
- `shape`, `scale`: shape and scale parameters, the latter defaulting to 1.
- `log`: logical; if TRUE, returns the log of the pdf.

Here are plots of the Weibull density function for a few different values of the parameters $k$ and $\lambda$:

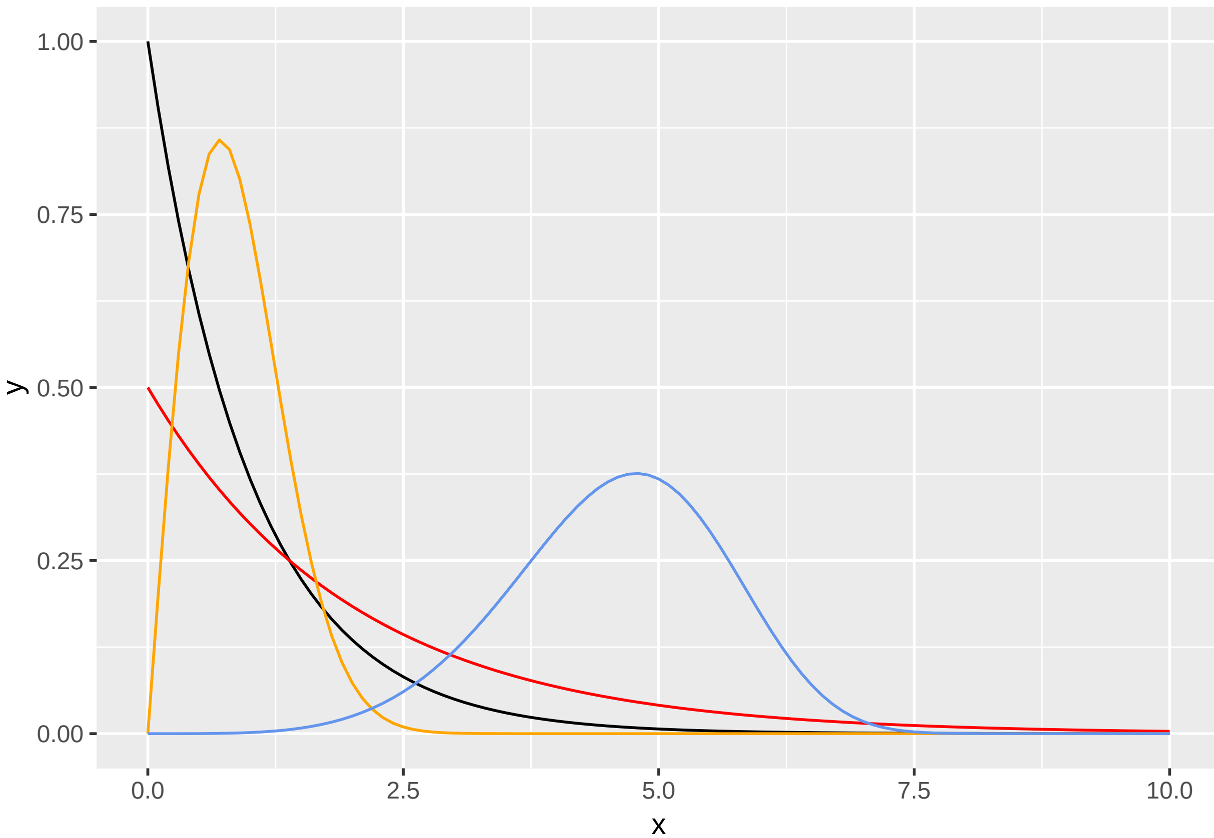
```r
library(tidyverse)
```

```
## -- Attaching packages -------------------------------------------------------------------------------

## v tibble  2.0.1     v purrr   0.3.0
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.3.1     v forcats 0.3.0

## Warning: package 'tibble' was built under R version 3.5.2

## Warning: package 'purrr' was built under R version 3.5.2

## -- Conflicts ---------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

```r
ggplot(data = data.frame(x = c(0, 10)), mapping = aes(x = x)) +
  stat_function(fun = dweibull, args = list(shape = 1, scale = 1)) +
  stat_function(fun = dweibull, args = list(shape = 1, scale = 2), color = "red") +
  stat_function(fun = dweibull, args = list(shape = 2, scale = 1), color = "orange") +
  stat_function(fun = dweibull, args = list(shape = 5, scale = 5), color = "cornflowerblue")
```




In this lab, you'll fit a Weibull distribution to model measurements of daily average wind speeds in meters/second at the site of a wind turbine in Minnesota over the course of 168 days from February 14 to August 1, 2010 (there were no data for July 2). The following R code reads the data in and makes a couple of initial plots:

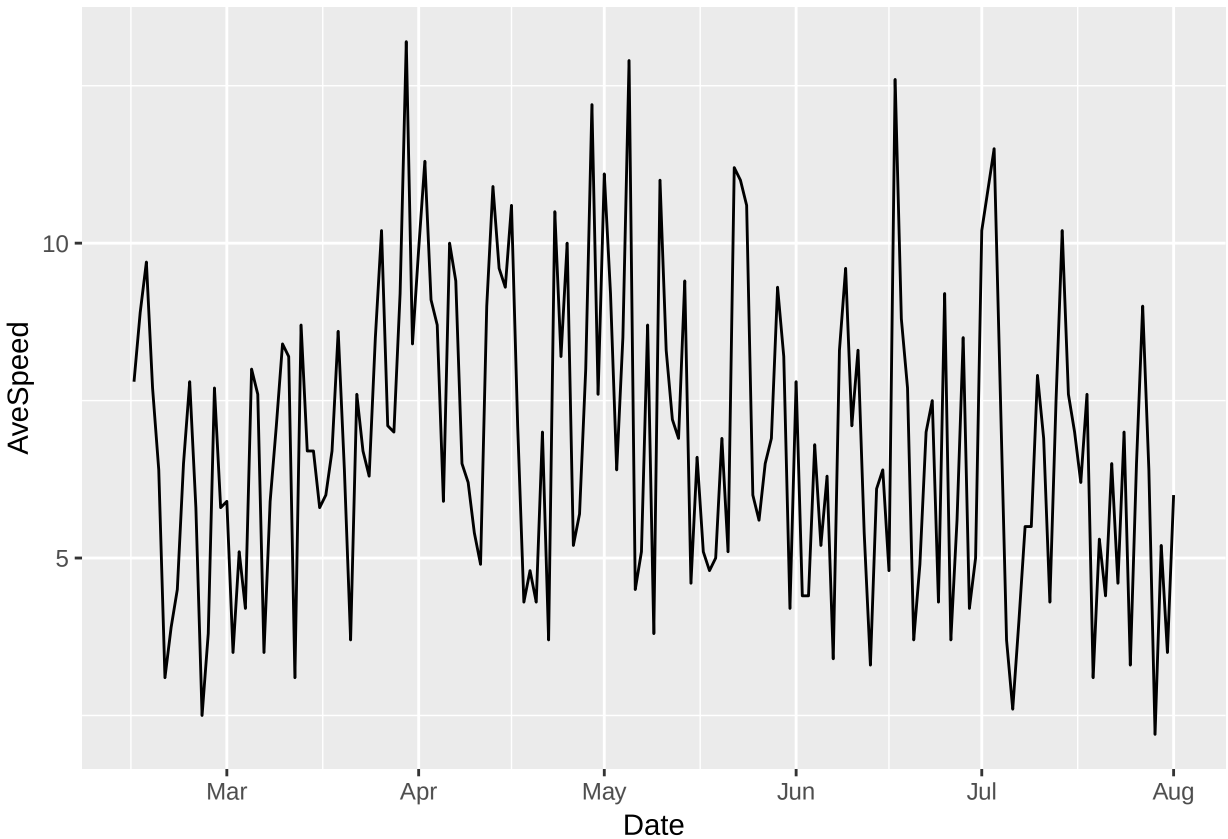
```
wind_speeds <- read_csv("http://www.evanlray.com/data/chihara_hesterberg/Turbine.csv") %>%
  mutate(Date = mdy(paste0(Date2010, " 2010")))
```

```
## Parsed with column specification:
## cols(
##   Date2010 = col_character(),
##   AveKW = col_double(),
##   AveSpeed = col_double(),
##   Production = col_double()
## )
```

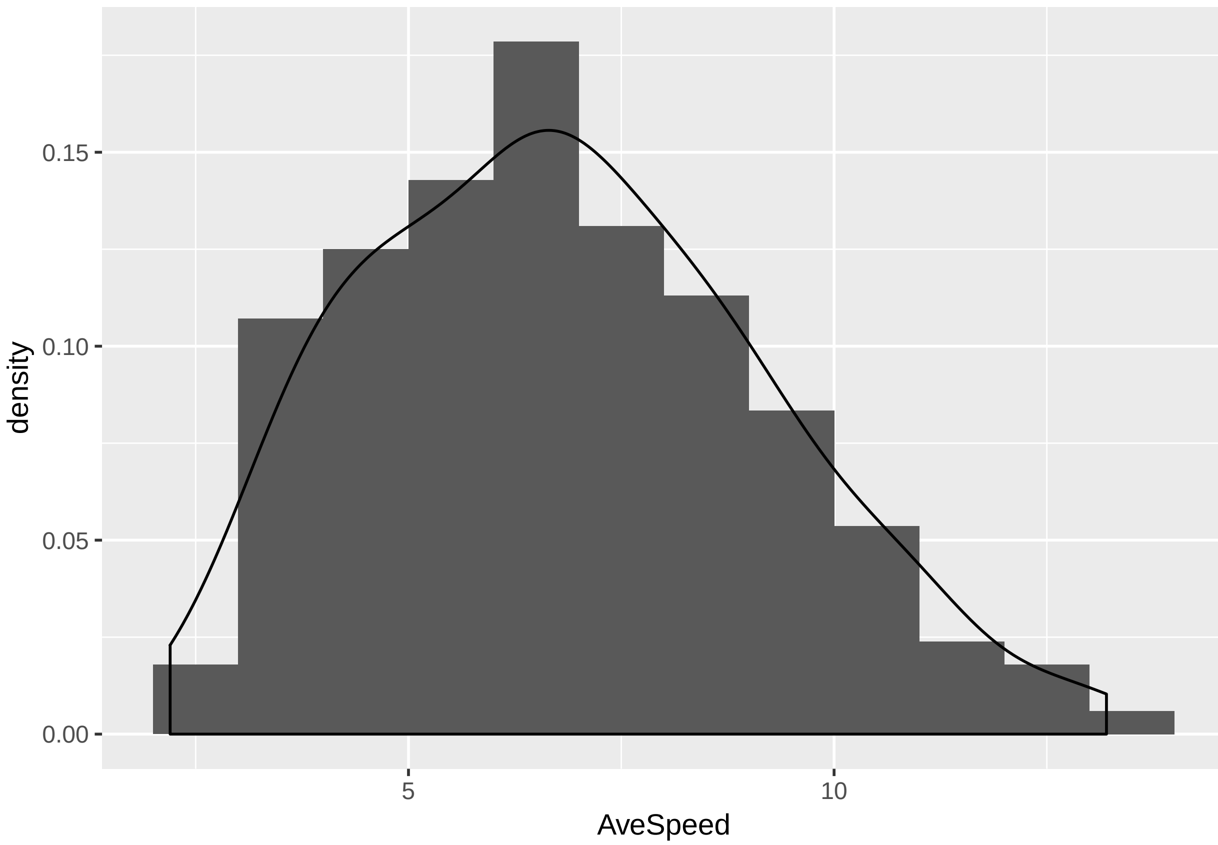
```
head(wind_speeds)
```

```
## # A tibble: 6 x 5
##   Date2010 AveKW AveSpeed Production Date
##   <chr>    <dbl>    <dbl>      <dbl> <date>
## 1 Feb 14   548.       7.8      13146 2010-02-14
## 2 Feb 15   776        8.9      18626 2010-02-15
## 3 Feb 16   944.       9.7      22667 2010-02-16
## 4 Feb 17   506.       7.7      12148 2010-02-17
## 5 Feb 18   323.       6.4       7742 2010-02-18
## 6 Feb 19    67.9      3.1       1585 2010-02-19
```

```r
ggplot(data = wind_speeds, mapping = aes(x = Date, y = AveSpeed)) +
  geom_line()
```




```r
ggplot(data = wind_speeds, mapping = aes(x = AveSpeed)) +
  geom_histogram(binwidth = 1, center = 0.5, mapping = aes(y = ..density..)) +
  geom_density()
```




Although data collected over time are basically never independent, let's model these observations as independent for the sake of having an example that's feasible to work with. So, we'll use the model

$X_i \overset{\text{iid}}{\sim} \text{Weibull}(k, \lambda)$,

where $X_i$ is the observed average wind speed for day $i$.

#### 1. Set up model definition in stan

I have set up a skeleton of the stan file, included in this repository. Edit that file now to add necessary declarations and model statements for this model to the data, parameters, and model blocks. The stan function to use for the Weibull distribution is called `weibull`.

#### 2. Perform estimation

You will need to load the rstan package, set up a list with the data for the stan model, compile the model, and call `optimizing` to obtain the maximum likelihood estimates of the model parameters.

```r
# Load the rstan package
library(rstan)
```

```
## Loading required package: StanHeaders
```

```
## Warning: package 'StanHeaders' was built under R version 3.5.2
```

```
## rstan (Version 2.18.2, GitRev: 2e1f913d3ca3)
```

```
## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
```

```
## rstan_options(auto_write = TRUE)
##
## Attaching package: 'rstan'
## The following object is masked from 'package:tidyr':
##
##     extract
```

```r
# Set up list with data Stan will need to know about
stan_data <- list(
  n = nrow(wind_speeds),
  x = wind_speeds$AveSpeed
)

# Compile the Stan model definition to an executable.  Takes a few seconds to run.
wind_model_compiled <- stan_model(file = "wind_model.stan")

# Call Stan to do optimization
wind_fit <- optimizing(wind_model_compiled,
  data = stan_data,
  seed = 8742,
  init = "random"
)
```

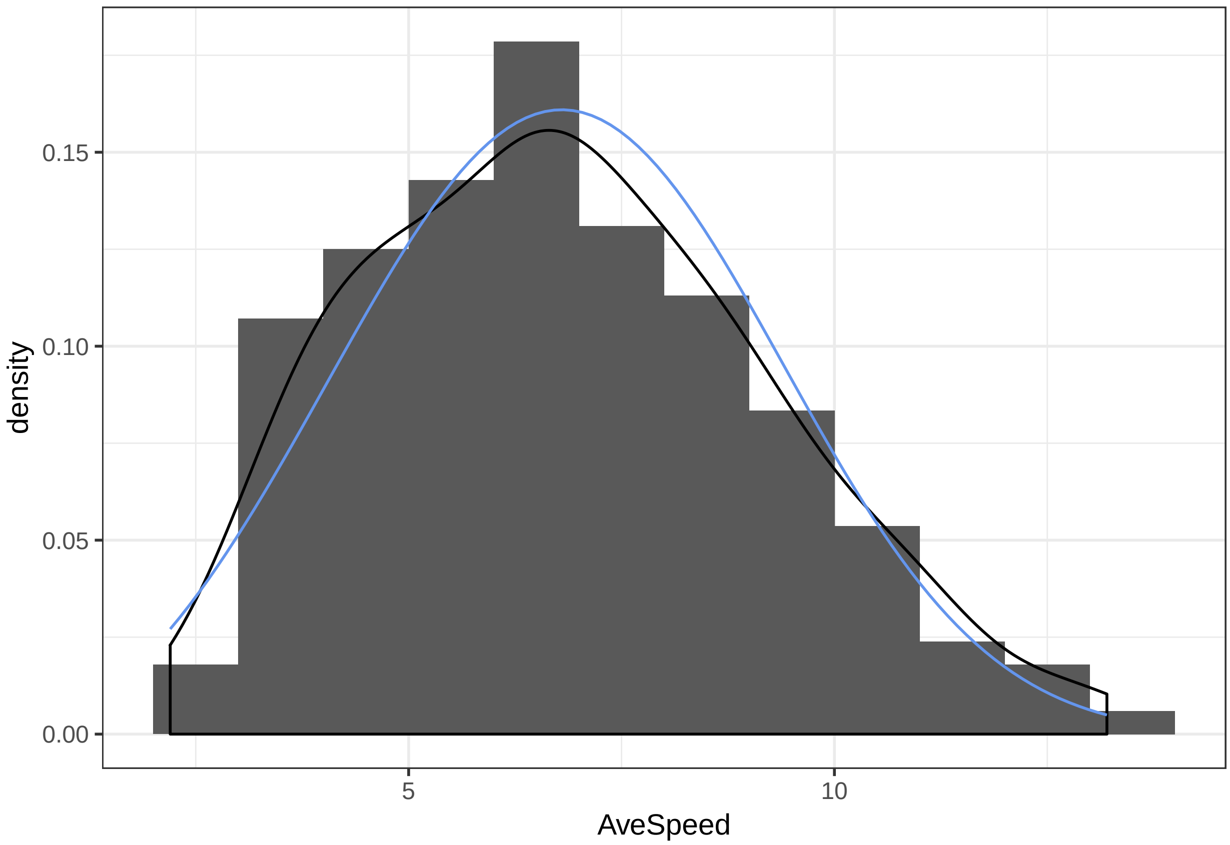
```r
# Here's a look at the return object, which is a list with 3 components:
#  * par is a named vector of parameter estimates
#  * value is the value of the log-likelihood at the maximum,
#    after dropping constants that don't involve the parameters
#  * return_code is 0 if everything went well in the optimization procedure,
#    otherwise an error code to be sad about
wind_fit
```

```
## $par
##        k   lambda
## 3.169321 7.661277
##
## $value
## [1] -380.0201
##
## $return_code
## [1] 0
```

### 3. Make a plot

Add a new layer to the plot below showing the pdf of the Weibull distribution corresponding to the maximum likelihood parameter estimates.

```r
ggplot(data = wind_speeds, mapping = aes(x = AveSpeed)) +
  geom_histogram(binwidth = 1, center = 0.5, mapping = aes(y = ..density..)) +
  geom_density() +
  stat_function(fun = dweibull, args = list(shape = wind_fit$par["k"], scale = wind_fit$par["lambda"]),
  theme_bw()
```


0.15
0.10
0.05
0.00
density
5
10
AveSpeed