Lab10

Lin XIAO

November 28, 2015

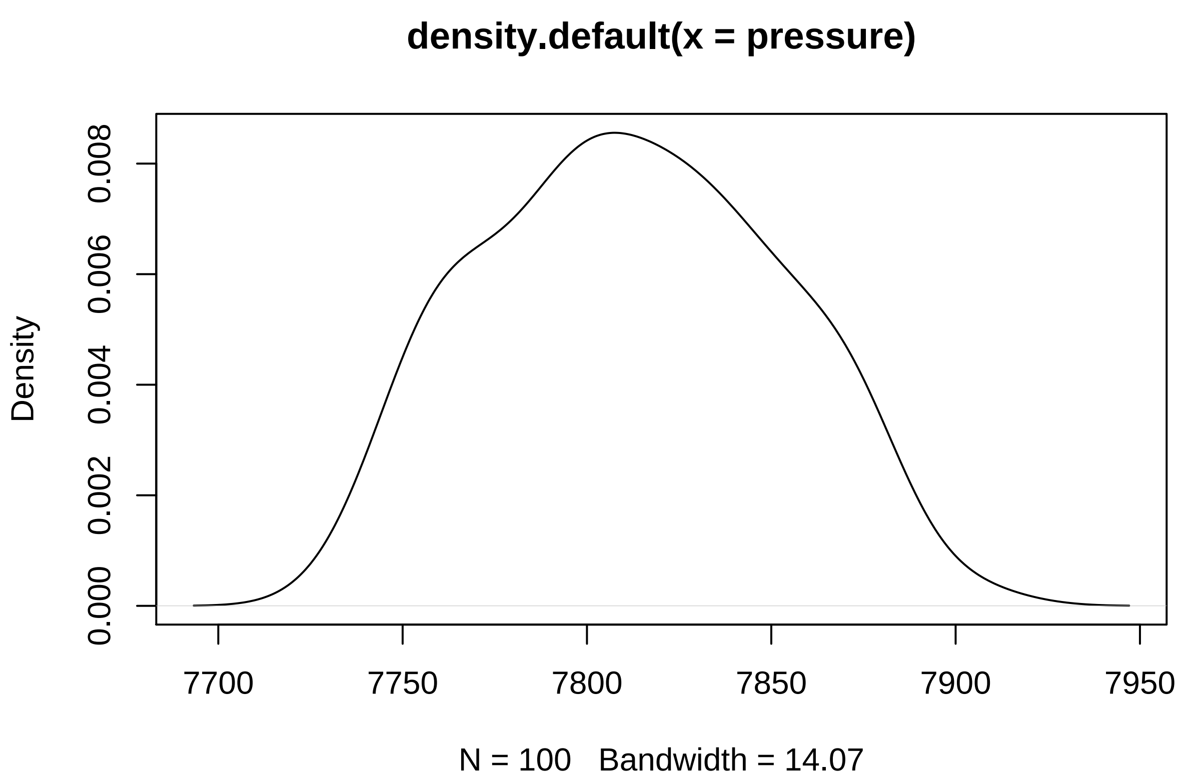
The mail goal of today is to apply the bootstrap to a linear regression model and use this to produce confidence intervals for the coefficient estimates. Start this lab by reading in the data file **Lab 10 Data.csv** into R.

1. Perform uni-variate and bi-variate data analysis on the variables in the model. Your response variable is Pressure and your input variable is Temperature. Fit a linear regression model and create three diagnostic plots: an independence plot, a QQ-plot, and a plot of residuals vs. temperature. Additionally produce a table of coefficient estimates with xtable.

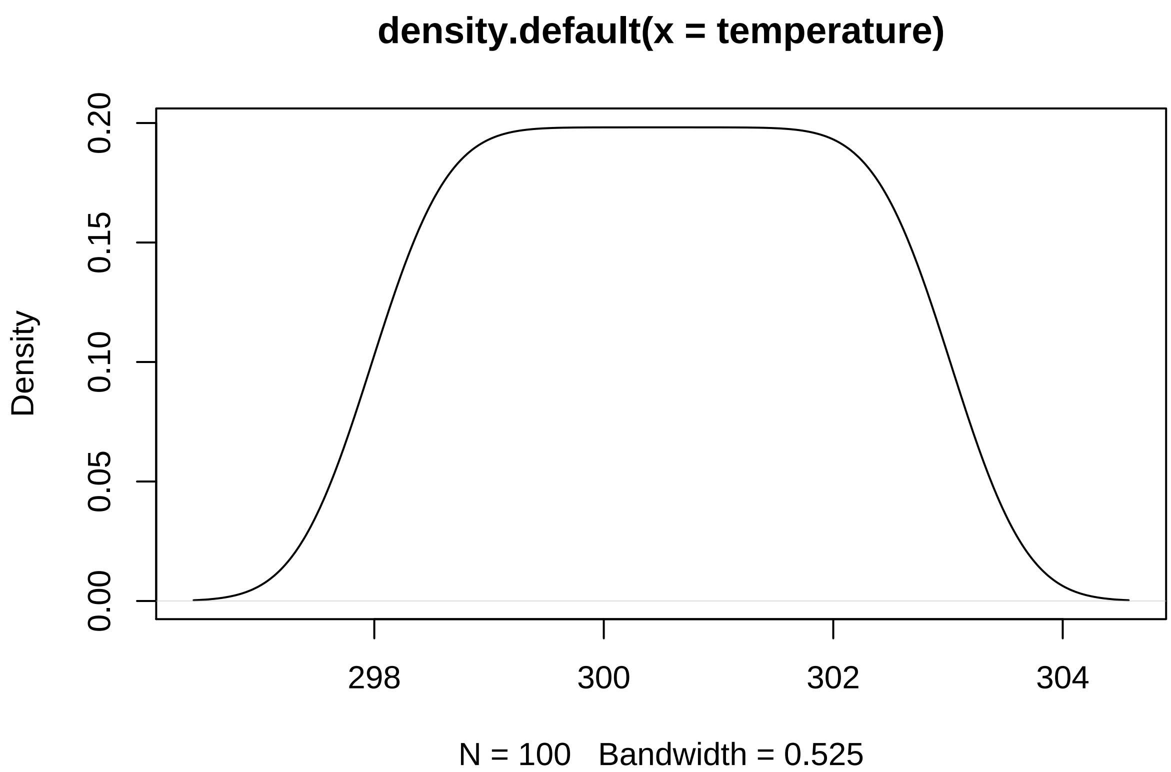
```r
library(xtable)
data <- read.csv("Lab+10+Data.csv", header = T)
attach(data)
```

```
## The following object is masked from package:datasets:
##
##     pressure
```

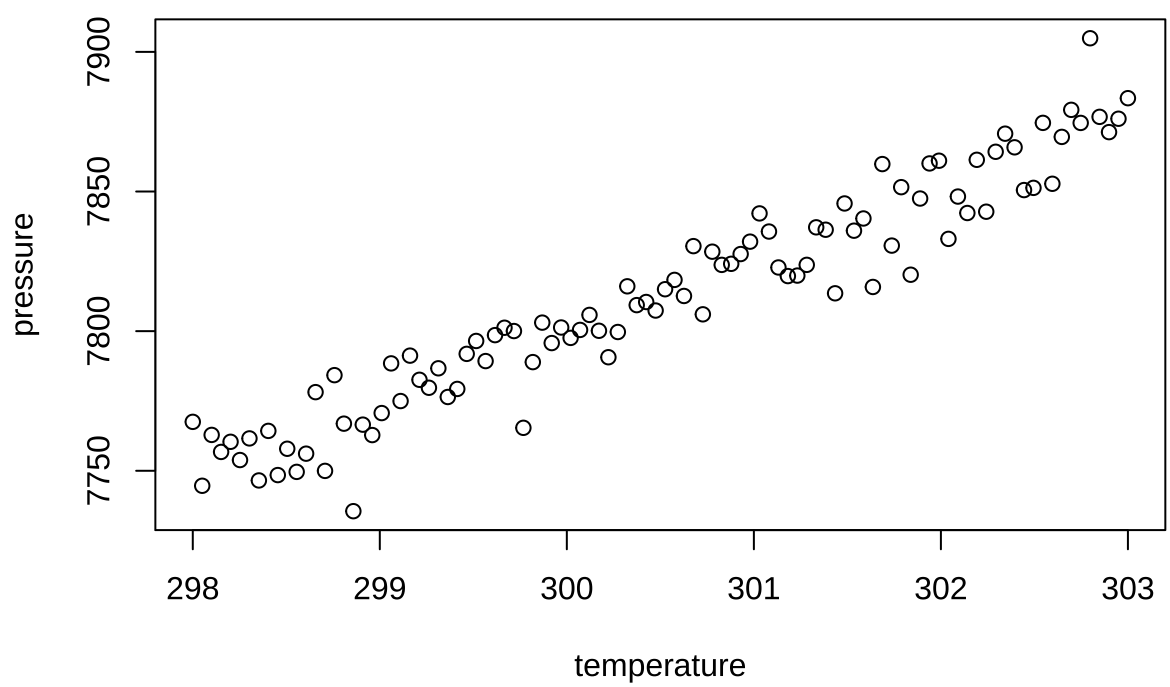
```r
# Analysis on univariates
plot(density(pressure))
```




```r
plot(density(temperature))
```

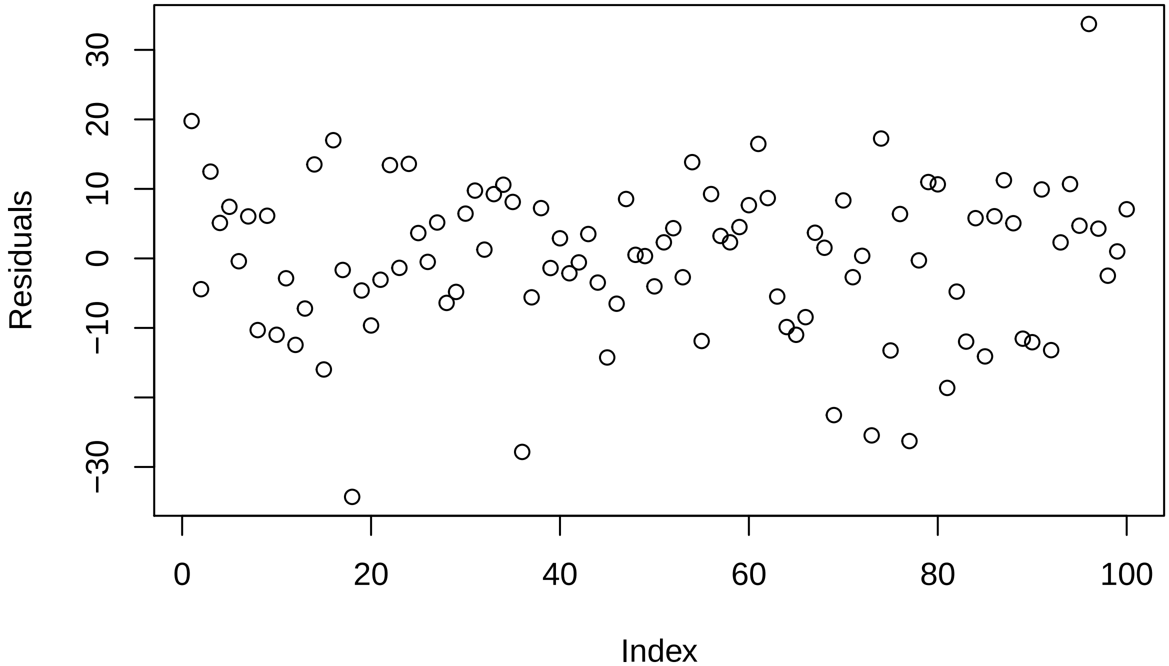



```r
# Analysis on bivariate
plot(temperature, pressure)
```

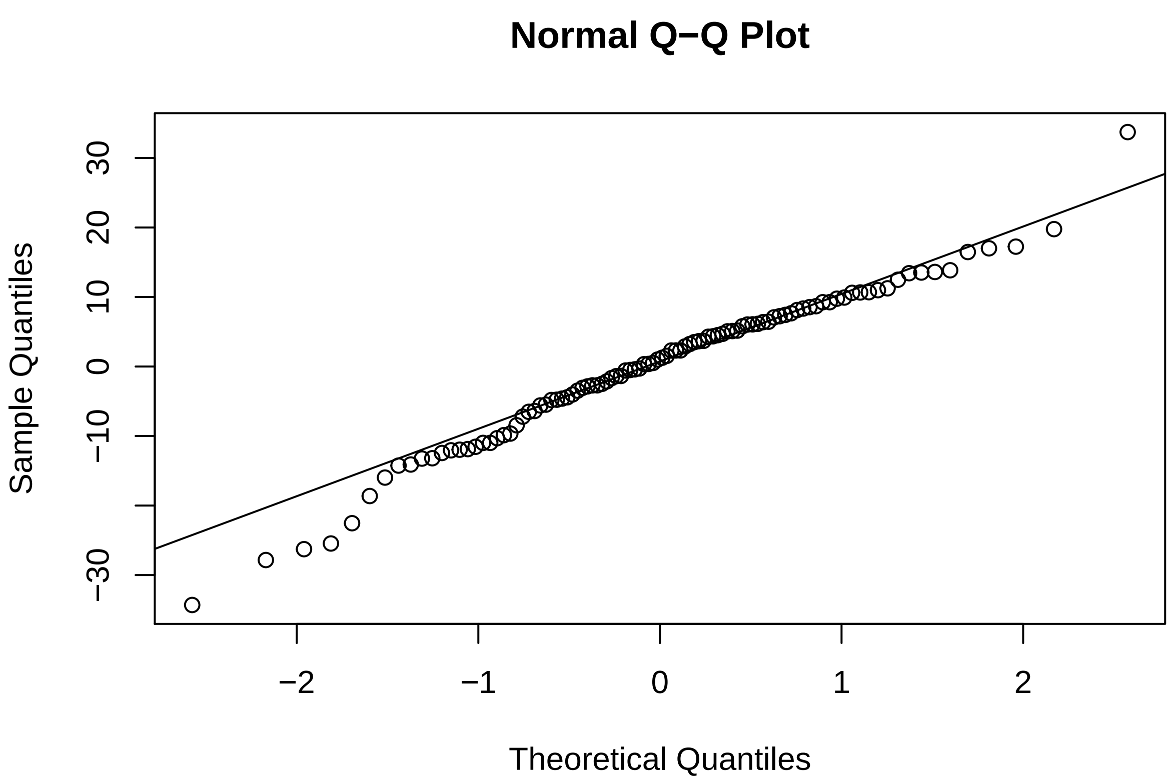



```r
# Fit a linear model
model <- pressure~temperature
fit1 <- lm(model)
```

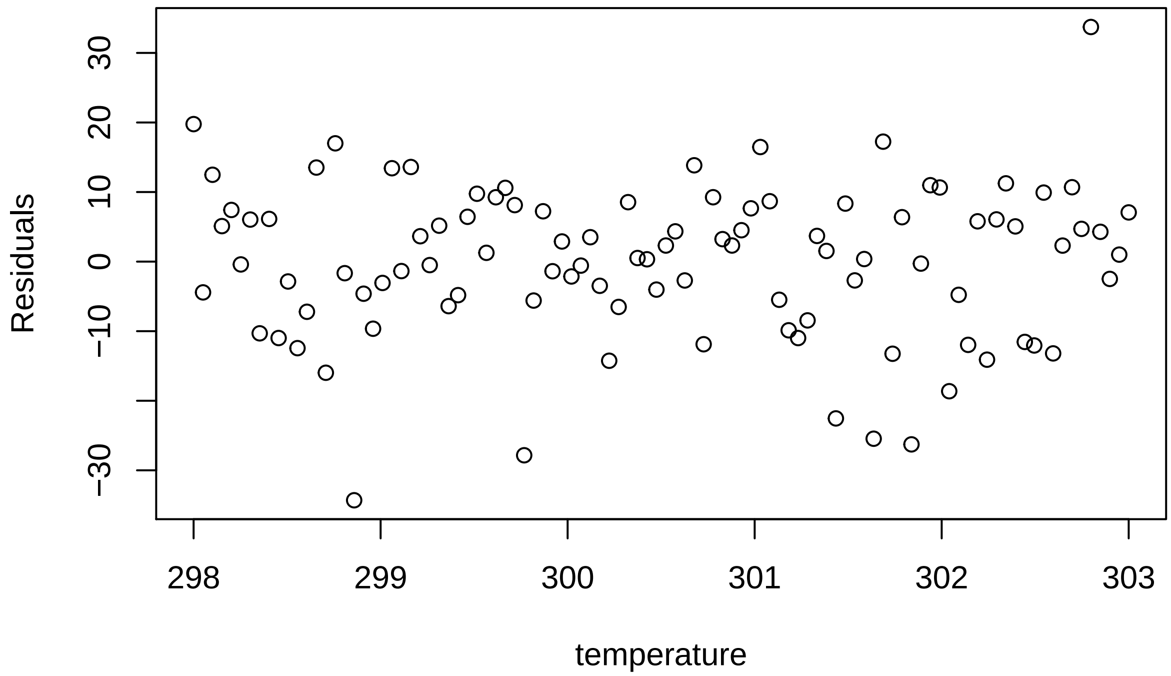
```r
# Independence plot
plot(residuals(fit1), ylab = "Residuals")
```




```r
# QQ-plot with a line
qqnorm(residuals(fit1))
qqline(residuals(fit1))
```



```r
# Plot of residuals vs. temperature
plot(temperature, residuals(fit1), ylab = "Residuals")
```




```r
# XTABLE
dat <- data.frame(summary(fit1)$coefficients)
print(xtable(dat, caption = "Coefficient estimates", digits = 4), comment=F)
```

	Estimate	Std..Error	t.value	Pr...t..
(Intercept)	83.3452	228.4621	0.3648	0.7160
temperature	25.7195	0.7603	33.8297	0.0000

Table 1: Coefficient estimates

We can see from the plots that the univariates are not skewed and there is a strong postive linear relationship between pressure and temperature.

2. Write a function that takes as its argument a data frame and returns a resampled 'synthetic' dataset of equal dimension.

```r
# Input: DF, a data frame
# Output: a resampled data frame of equal dimension
resample <- function(DF){
  n <- dim(DF)[1]
  newRows <- sample(n, replace = T, size = n)
  newDF <- DF[newRows, ]
  return(newDF)
}
```

3. Write a function that takes as its input a string model specification and a dataframe and produces a fitted model object from lm() with a resampled data frame using the function you have written above.

```
# Input: a string model specification and a data frame
# Output: a fitted model object
model.fit <- function(model, DF){
  modelfit <- lm(model, data = resample(DF))
  return(modelfit)
}
```

4. Write a function that takes as its inputs a data frame, a string model and a natural number representing how many bootstrap replications to run and returns the coefficients from each replication using the function you have written above.

```
# Estimate the bootsrapped coefficients
# Input: a data frame, a string model and a natural number
# representing how many bootstrap replications to run
# Output: coefficients from each replication
bootstrap.coefs <- function(B, DF, model){
  boot.coefs <- replicate(B, coef(model.fit(model, DF)))
  return(boot.coefs)
}
```

5. Write a function that uses all three of the functions you have written above, and takes four arguments: a count of replications, type I error rate, a data frame, and a string model to produce a bootstrapped confidence interval for the regression coefficients. Run this function and provide (using xtable) the confidence interval for your regression coefficients. What do you conclude?

```
# Estimate the bootsrapped coefficients confidence intervals
# Inputs: a count of replications, type I error rate,
# a data frame, and a string model
# Output: confidence interval for regression coefficients
bootstrap.coef.ci <- function(B, alpha, DF, model){
  original_coef <- coef(lm(model, data = DF))
  coefs <- bootstrap.coefs(B, DF, model)
  low.q <- apply(coefs, 1, quantile, probs = alpha/2)
  high.q <- apply(coefs, 1, quantile, probs = 1-alpha/2)

  # Calculate the confidence intervals
  low.cis <- 2*original_coef - high.q
  high.cis <- 2*original_coef - low.q
  cis <- rbind(low.cis, high.cis)
  rownames(cis) <- as.character(c(alpha/2, 1-alpha/2))
  return(cis)
}


set.seed(123)
results <- bootstrap.coef.ci(B = 1000, alpha = 0.05, DF = data, model = model)
print(xtable(results, caption = "Bootstrapped Confidence Intervals",
             digits = 4), comment = F)
```

Conclusion: From these two tables we can see that the coefficient estimates of the original linear model are each within the range of confidence interval. That means bootstrap is considered useful for producing confidence intervals for the coefficients estimates.

	(Intercept)	temperature
0.025	-368.9939	24.0992
0.975	570.4673	27.2312

Table 2: Bootstrapped Confidence Intervals