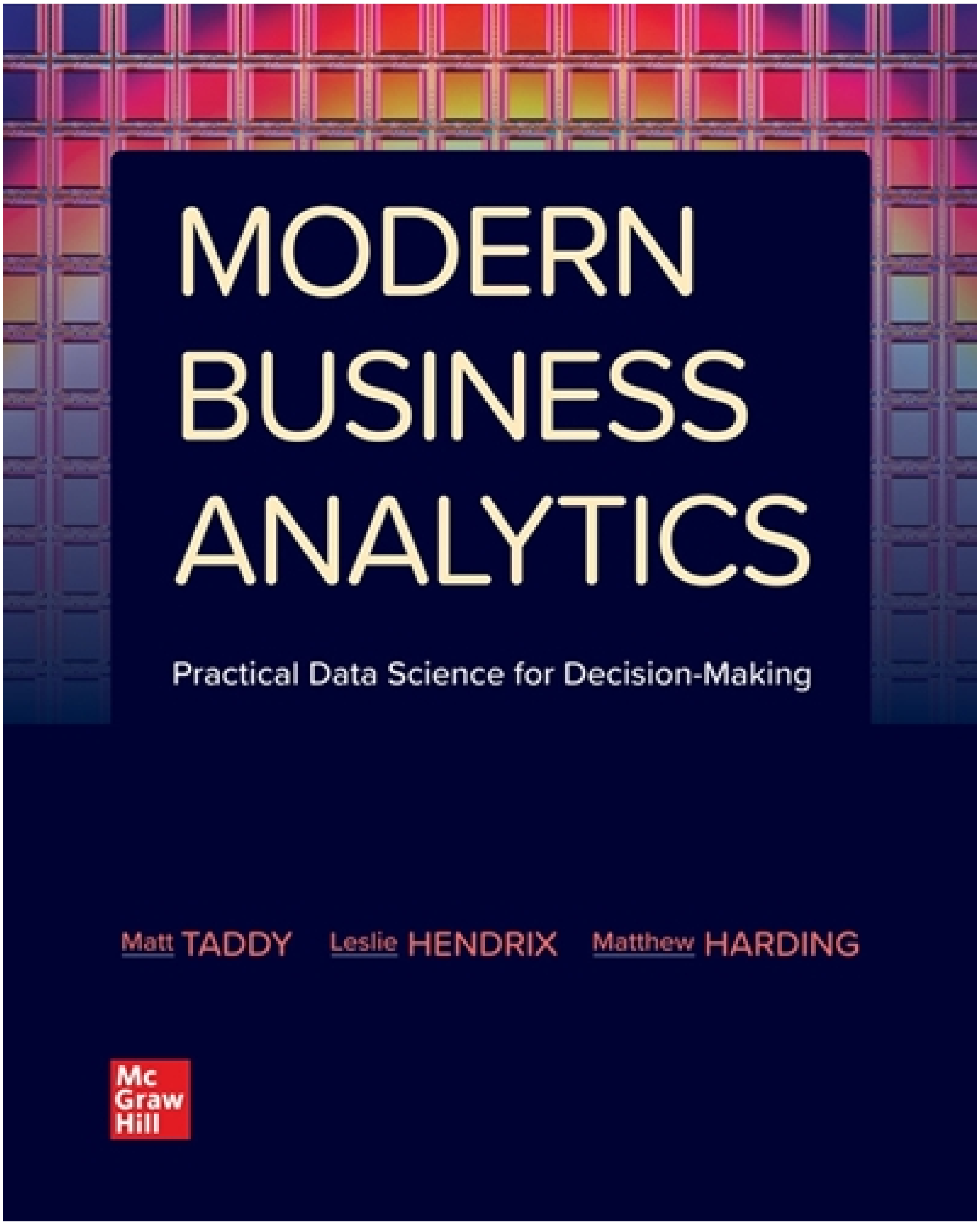
Solutions for Modern Business Analytics 1st Edition by Taddy


MODERN
BUSINESS
ANALYTICS
Practical Data Science for Decision-Making
Matt TADDY Leslie HENDRIX Matthew HARDING
Mc Graw Hill

Solutions

Chapter 2

Uncertainty Quantification

The data in `ames2009.csv` consist of information that the local government in Ames, Iowa, uses to assess home values. These data were compiled from 2006 to 2010 by De Cock (2011) and contain 2930 observations on 79 variables describing properties in Ames and their observed sale price. For this problem, we will use a subset of the Ames data contained in `ames2009.csv`.

<Ames2009.csv>

Problem 2.1
a. What is the average sales price?
178368 (+/-0.1)

b. What is the standard error of the mean? (Note: If you use intermediate calculations, keep at least two decimal places for each and report your answer to three decimal places.)
2506.944 (+/-0.2)

c. Use the mean and standard error to calculate a 95% confidence interval for the mean unconditional sales price. Use 1.96 for the critical value. What is the lower bound and upper bound? (Note: If you use intermediate calculations, keep at least two decimal places for each and report your answer to one decimal place.)

Lower bound: 173454.4 (+/-0.1)
Upper bound: 183281.6 (+/-0.1)

Explanation/Solution

The following code will call in the data if it is in your working directory. Note the `strings=T` argument to treat character data as a factor in R.

```
ames <- read.csv("Ames2009.csv", strings=T) #call in data
```

a. The following code can be used to calculate the average sales price:

```
(xbar <- mean(ames$SalePrice)) #mean
```

b. The following code can be used to calculate the standard error of the mean:

```
(muSE <- sd(ames$SalePrice)/sqrt(nrow(ames))) #SE of the mean
```

c. The following code can be used to calculate the 95% confidence interval for the mean unconditional sales price:

```
xbar + c(-1,1)*1.96*muSE #95% CI
```

Problem 2.2 Regress the `log(SalePrice)` onto all variables except for `Neighborhood`. The following code will run the regression, assuming you called the data “ames”.

```
amesFit <- glm(log(SalePrice) ~ .-Neighborhood, data=ames)
```

Which of the following regression coefficients are significant when you control for a 5% false discovery rate? Choose all that apply. (Note: You can use the code below from the text to find the cutoff. You may select more than one answer.)

```
pvals <- summary(amesFit)$coef[-1,"Pr(>|t|)"]
fdr_cut <- function(pvals, q){
       pvals <- pvals[!is.na(pvals)]
       N <- length(pvals)
       k <- rank(pvals, ties.method="min")
       max(pvals[ pvals<= (q*k/N) ])
       }
cutoff5 <- fdr_cut(pvals,q=.05)
```

a. log.Lot.Area
b. Lot.Config
c. Bldg.Type
d. Overall.Qual
e. Overall.Cond
f. Year.Built
g. Central.Air
h. Electrical
i. Gr.Liv.Area
j. Full.Bath
k. Half.Bath
l. Bedroom.AbvGr
m. Kitchen.AbvGr
n. TotRms.AbvGrd

Explanation/Solution

The following code prints which regression coefficients are significant when you control for a 5% false discovery rate:

```
print(cutoff5) #FDR cut-off
which(pvals<=cutoff5) #find predictors with p-values below the cutoff
```

Problem 2.3 Regress the `log(SalePrice)` onto all variables except for `Neighborhood`.

a. What is the lower and upper bound of the 95% confidence interval for the effect of having central air on the expected log sale price? Note: Use the values output from glm and 1.96 for the critical value. Report your answer to four decimal places and carry as many decimal places as possible in intermediate calculations.

Lower bound: 0.0847 (+/-0.001)
Upper bound: 0.1795 (+/-0.001)

b. What is the lower and upper bound of the 95% confidence interval for the effect of having central air on the expected log sale price *using a bootstrap*? Note that the coefficient for the effect of having center air is `Central.AirY`. Note: Use the boot function from the boot library with 2,000 bootstrap samples. Before running the bootstrap, set the seed to 1. Report your answer to four decimal places and carry as many decimal places as possible in intermediate calculations.

The following function will extract the coefficients to feed to the boot function.

```
getBeta <- function(data, obs, var){
    fit <- glm(log(SalePrice) ~ .-Neighborhood, data=data[obs,])
    return(fit$coef[var])
    }
```

Lower bound: 0.0585 (+/-0.001)
Upper bound: 0.2162 (+/-0.001)

c. Which confidence interval is wider?

a. glm
b. bootstrap

Explanation/Solution

The following code will run the regression, assuming you called the data “ames”.

```
amesFit <- glm(log(SalePrice) ~ .-Neighborhood, data=ames)
```

a. The following code can be used to calculate the 95% confidence interval for the effect of having central air on the expected log sale price:

```
( bstats <- summary(amesFit)$coef["Central.AirY",] ) #coefficient
for Central Air
bstats["Estimate"] + c(-1,1)*1.96*bstats["Std. Error"] #95% CI
```

b. The following code can be used to use a bootstrap to calculate the 95% confidence interval:

```
library(parallel)
library(boot)
set.seed(1)
( betaBoot <- boot(ames, getBeta, 2000, var="Central.AirY",
       parallel="snow", ncpus=detectCores()) )
quantile(betaBoot$t, c(.025, .975))
```

c. The bootstrap interval is wider (the SE is 0.02 for the standard method and 0.04 for the bootstrap).

Problem 2.4. Regress the `log(SalePrice)` onto all variables except for `Neighborhood`.

a. Run the code below and feed these objects to the boot function. When using a block bootstrap, what is the lower and upper bound of the 95% confidence interval for the coefficient on central air while allowing for dependence in sales prices within neighborhoods? Note: Use 2,000 bootstrap samples. Before running the bootstrap, set the seed to 1. Report your answer to four decimal places.

```
byNBHD <- split(ames, ames$Neighborhood)
getBetaBlock <- function(data, ids, var){
    data <- do.call("rbind",data[ids])
    fit <- glm(log(SalePrice) ~ .-Neighborhood, data=data)
    return(fit$coef[var])
}
```

Lower bound: 0.0220 (+/-0.01)
Upper bound: 0.2031 (+/-0.01)

b. Use the `sandwich` package to obtain a 95% confidence interval for the coefficient on central air while allowing for dependence in sales prices within neighborhoods. Use 1.96 for the critical value. What is the lower and upper bound of this CI? Report your answer to *four decimal places*.

Upper bound: 0.2135 (+/-0.01)
Lower bound: 0.0507 (+/-0.01)

c. Use the results from the block bootstrap to calculate a bias-corrected 95% confidence interval for the multiplicative effect of central heating on the expected sale price. What is the lower and upper bound of this CI? Report your answer to *four decimal places*.

Upper bound: 1.2603 (+/-0.01)
Lower bound: 1.0574 (+/-0.01)

Explanation/Solution

The following code will run the regression, assuming you called the data "ames".

```
amesFit <- glm(log(SalePrice) ~ .-Neighborhood, data=ames)
```

a. The following code can be used to determine the confidence interval using a block bootstrap:

```
library(parallel)
library(boot)
```

```
set.seed(1)
( betaBootB <- boot(byNBHD, getBetaBlock, 2000, var="Central.AirY",
parallel="snow", ncpus=detectCores()) )
quantile(betaBootB$t, c(.025, .975))
```

Note that, even with the same seed, you may get different results for different operating systems and versions of R. You should at least be able to replicated your own results by running the same code twice on your own machine.

b. The following code uses the sandwich package to obtain a 95% confidence interval for the coefficient on central air while allowing for dependence in sales prices within neighborhoods:

```
library(sandwich)
library(lmtest)
Vblock <- vcovCL(amesFit, cluster=ames$Neighborhood)
clstats <- coeftest(amesFit, vcov = Vblock)["Central.AirY",]
round(clstats, 5)
clstats["Estimate"] + c(-1,1)*1.96*clstats["Std. Error"]
```

c. This multiplicative effect is the exponentiated coefficient. The exponentiation is a nonlinear transformation, and the distribution of this transformation will not be equal to the transformation of the raw coefficient distribution (i.e., `exp(beta.hat)` is a biased estimate for true `exp(beta)`). So, we should use the bias corrected bootstrap to obtain the 95% CI:

```
quantile(2*exp(betaBootB$t0) - exp(betaBootB$t), c(.025, .975))
```

If you got different results for the bookstrap in part a, your answer might be slightly different than the answer key shows.