STAT HW1

2025-04-01

## Question 1: Patient Satisfaction Data

### a) Exploratory Data Analysis

# Load the patient satisfaction data with explicit column classes  
raw\_sat <- read.table("pat\_stat.txt",   
 header=TRUE,   
 nrows=46,  
 colClasses="numeric")  
  
# Check the data structure  
head(raw\_sat)

## pat\_sat pat\_age severity anxiety  
## 1 48 50 51 2.3  
## 2 57 36 46 2.3  
## 3 66 40 48 2.2  
## 4 70 41 44 1.8  
## 5 89 28 43 1.8  
## 6 36 49 54 2.9

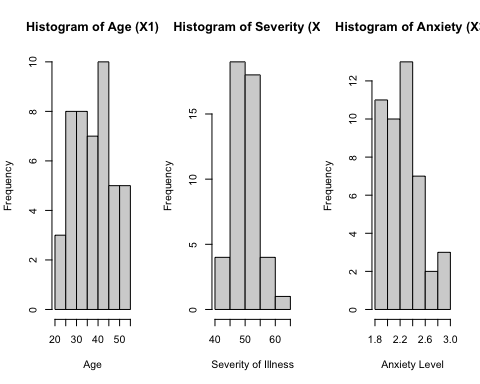
str(raw\_sat)

## 'data.frame': 46 obs. of 4 variables:  
## $ pat\_sat : num 48 57 66 70 89 36 46 54 26 77 ...  
## $ pat\_age : num 50 36 40 41 28 49 42 45 52 29 ...  
## $ severity: num 51 46 48 44 43 54 50 48 62 50 ...  
## $ anxiety : num 2.3 2.3 2.2 1.8 1.8 2.9 2.2 2.4 2.9 2.1 ...

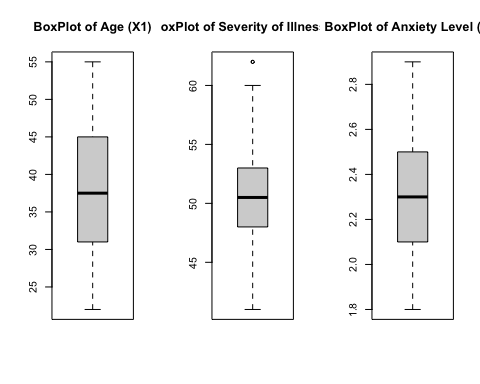
# Create a new data frame with proper column names  
pat\_sat <- data.frame(  
 Y = raw\_sat$pat\_sat, # Using the correct variable name raw\_sat  
 X1 = raw\_sat$pat\_age,  
 X2 = raw\_sat$severity,  
 X3 = raw\_sat$anxiety  
)  
  
# Verify the structure is now correct  
str(pat\_sat)

## 'data.frame': 46 obs. of 4 variables:  
## $ Y : num 48 57 66 70 89 36 46 54 26 77 ...  
## $ X1: num 50 36 40 41 28 49 42 45 52 29 ...  
## $ X2: num 51 46 48 44 43 54 50 48 62 50 ...  
## $ X3: num 2.3 2.3 2.2 1.8 1.8 2.9 2.2 2.4 2.9 2.1 ...

# Histograms  
par(mfrow=c(1,3))  
hist(pat\_sat$X1, main="Histogram of Age (X1)", xlab="Age")  
hist(pat\_sat$X2, main="Histogram of Severity (X2)", xlab="Severity of Illness")  
hist(pat\_sat$X3, main="Histogram of Anxiety (X3)", xlab="Anxiety Level")



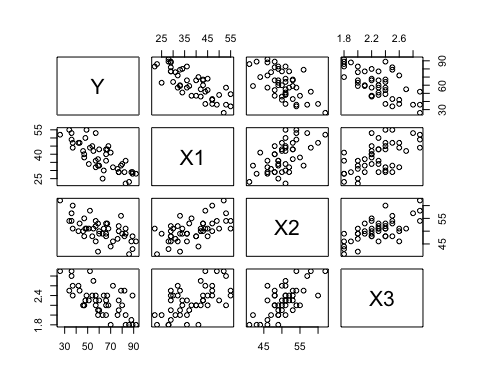
# Box plots for predictor variables  
par(mfrow=c(1,3))  
boxplot(pat\_sat$X1, main="BoxPlot of Age (X1)")  
boxplot(pat\_sat$X2, main="BoxPlot of Severity of Illness (X2)")  
boxplot(pat\_sat$X3, main="BoxPlot of Anxiety Level (X3)")



# Summary statistics  
summary(pat\_sat$X1) # Age  
summary(pat\_sat$X2) # Severity of illness  
summary(pat\_sat$X3) # Anxiety level

### b) Scatter Plot matrix and Correlation Matrix

# Scatter plot matrix  
pairs(pat\_sat)



# Correlation matrix  
cor(pat\_sat)

## Y X1 X2 X3  
## Y 1.0000000 -0.7867555 -0.6029417 -0.6445910  
## X1 -0.7867555 1.0000000 0.5679505 0.5696775  
## X2 -0.6029417 0.5679505 1.0000000 0.6705287  
## X3 -0.6445910 0.5696775 0.6705287 1.0000000

### c) Multiple Linear Regression Model

# Create the model  
model <- lm(Y ~ X1 + X2 + X3, data=pat\_sat)  
  
# Now check the summary  
summary(model)

##   
## Call:  
## lm(formula = Y ~ X1 + X2 + X3, data = pat\_sat)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -18.3524 -6.4230 0.5196 8.3715 17.1601   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 158.4913 18.1259 8.744 5.26e-11 \*\*\*  
## X1 -1.1416 0.2148 -5.315 3.81e-06 \*\*\*  
## X2 -0.4420 0.4920 -0.898 0.3741   
## X3 -13.4702 7.0997 -1.897 0.0647 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 10.06 on 42 degrees of freedom  
## Multiple R-squared: 0.6822, Adjusted R-squared: 0.6595   
## F-statistic: 30.05 on 3 and 42 DF, p-value: 1.542e-10

### d) Overall Model Significance Test

# F-test results are available in the model summary  
summary(model)

##   
## Call:  
## lm(formula = Y ~ X1 + X2 + X3, data = pat\_sat)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -18.3524 -6.4230 0.5196 8.3715 17.1601   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 158.4913 18.1259 8.744 5.26e-11 \*\*\*  
## X1 -1.1416 0.2148 -5.315 3.81e-06 \*\*\*  
## X2 -0.4420 0.4920 -0.898 0.3741   
## X3 -13.4702 7.0997 -1.897 0.0647 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 10.06 on 42 degrees of freedom  
## Multiple R-squared: 0.6822, Adjusted R-squared: 0.6595   
## F-statistic: 30.05 on 3 and 42 DF, p-value: 1.542e-10

### e) Confidence Interval for β₁

# 90% confidence interval for β₁  
confint(model, level=0.9)

## 5 % 95 %  
## (Intercept) 128.004370 188.9781330  
## X1 -1.502893 -0.7803305  
## X2 -1.269467 0.3854587  
## X3 -25.411454 -1.5288719

### f) Coefficient of Multiple Determination

# R² value from model summary  
summary(model)$r.squared

## [1] 0.6821943

### g) Prediction

# New data for prediction  
new\_data <- data.frame(X1=35, X2=45, X3=2.2)  
  
# Predict with prediction intervals  
predicted\_values <- predict(model, newdata=new\_data, interval="prediction", level=0.9)   
predicted\_values

## fit lwr upr  
## 1 69.01029 51.50965 86.51092

### h) Model Selection

# Full model  
full\_model <- lm(Y ~ X1 + X2 + X3, data=pat\_sat)  
  
# Forward selection  
null\_model <- lm(Y ~ 1, data=pat\_sat)   
forward\_model <- step(null\_model, scope=list(lower=null\_model, upper=full\_model),   
 direction="forward", trace=1)

## Start: AIC=262.92  
## Y ~ 1  
##   
## Df Sum of Sq RSS AIC  
## + X1 1 8275.4 5093.9 220.53  
## + X3 1 5554.9 7814.4 240.21  
## + X2 1 4860.3 8509.0 244.13  
## <none> 13369.3 262.92  
##   
## Step: AIC=220.53  
## Y ~ X1  
##   
## Df Sum of Sq RSS AIC  
## + X3 1 763.42 4330.5 215.06  
## + X2 1 480.92 4613.0 217.97  
## <none> 5093.9 220.53  
##   
## Step: AIC=215.06  
## Y ~ X1 + X3  
##   
## Df Sum of Sq RSS AIC  
## <none> 4330.5 215.06  
## + X2 1 81.659 4248.8 216.19

# Backward selection  
backward\_model <- step(full\_model, direction="backward", trace=1)

## Start: AIC=216.18  
## Y ~ X1 + X2 + X3  
##   
## Df Sum of Sq RSS AIC  
## - X2 1 81.66 4330.5 215.06  
## <none> 4248.8 216.19  
## - X3 1 364.16 4613.0 217.97  
## - X1 1 2857.55 7106.4 237.84  
##   
## Step: AIC=215.06  
## Y ~ X1 + X3  
##   
## Df Sum of Sq RSS AIC  
## <none> 4330.5 215.06  
## - X3 1 763.4 5093.9 220.53  
## - X1 1 3483.9 7814.4 240.21

# Compare AIC values  
AIC(forward\_model)

## [1] 347.603

AIC(backward\_model)

## [1] 347.603

## Question 2: Muscle Mass Data

# Try the simplest approach - skip the header line  
mmass\_data <- read.table("muscle\_mass.txt", header=FALSE, skip=1, nrows=60)  
names(mmass\_data) <- c("Y", "X") # Assign column names  
  
  
# Make sure everything is numeric  
mmass\_data$Y <- as.numeric(as.character(mmass\_data$Y))  
mmass\_data$X <- as.numeric(as.character(mmass\_data$X))  
  
# Check that it worked  
head(mmass\_data)

## Y X  
## 1 106 43  
## 2 106 41  
## 3 97 47  
## 4 113 46  
## 5 96 45  
## 6 119 41

str(mmass\_data)

## 'data.frame': 60 obs. of 2 variables:  
## $ Y: num 106 106 97 113 96 119 92 112 92 102 ...  
## $ X: num 43 41 47 46 45 41 47 41 48 48 ...

### a) Correlation between age and muscle mass

# Calculate correlation  
correlation <- cor(mmass\_data$X, mmass\_data$Y)  
correlation

## [1] -0.866064

### b) First-order regression model

# Fit first-order regression model  
model1 <- lm(Y ~ X, data=mmass\_data)  
summary(model1)

##   
## Call:  
## lm(formula = Y ~ X, data = mmass\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -16.1368 -6.1968 -0.5969 6.7607 23.4731   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 156.3466 5.5123 28.36 <2e-16 \*\*\*  
## X -1.1900 0.0902 -13.19 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 8.173 on 58 degrees of freedom  
## Multiple R-squared: 0.7501, Adjusted R-squared: 0.7458   
## F-statistic: 174.1 on 1 and 58 DF, p-value: < 2.2e-16

# R-squared value  
r\_squared1 <- summary(model1)$r.squared  
r\_squared1

## [1] 0.7500668

# Plot the data and fitted regression line  
plot(mmass\_data$X, mmass\_data$Y, main="First-Order Regression Model",  
 xlab="Age", ylab="Muscle Mass", pch=16)  
abline(model1, col="red")



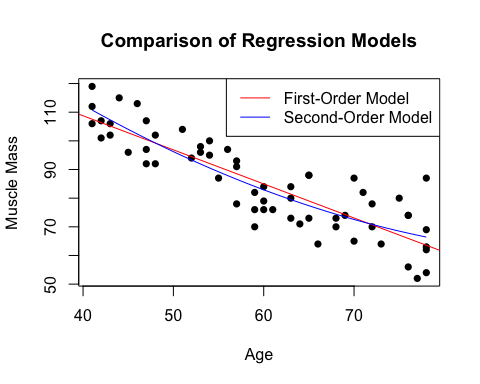
### c) Second-order regression model

# Fit second-order regression model  
model2 <- lm(Y ~ X + I(X^2), data=mmass\_data)  
summary(model2)

##   
## Call:  
## lm(formula = Y ~ X + I(X^2), data = mmass\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -15.086 -6.154 -1.088 6.220 20.578   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 207.349608 29.225118 7.095 2.21e-09 \*\*\*  
## X -2.964323 1.003031 -2.955 0.00453 \*\*   
## I(X^2) 0.014840 0.008357 1.776 0.08109 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 8.026 on 57 degrees of freedom  
## Multiple R-squared: 0.7632, Adjusted R-squared: 0.7549   
## F-statistic: 91.84 on 2 and 57 DF, p-value: < 2.2e-16

### d) Comparing regression models

# Plot the data  
plot(mmass\_data$X, mmass\_data$Y, main="Comparison of Regression Models",  
 xlab="Age", ylab="Muscle Mass", pch=16)  
  
# Add first-order model (red line)  
abline(model1, col="red")  
  
# Add second-order model (blue curve)  
x\_values <- seq(min(mmass\_data$X), max(mmass\_data$X), length.out=100)  
y\_pred <- predict(model2, newdata=data.frame(X=x\_values))  
lines(x\_values, y\_pred, col="blue")  
  
# Add legend  
legend("topright", legend=c("First-Order Model", "Second-Order Model"),   
 col=c("red", "blue"), lty=1)



### e) Test significance of second-order model

# Test of overall significance for second-order model  
anova(model2)

## Analysis of Variance Table  
##   
## Response: Y  
## Df Sum Sq Mean Sq F value Pr(>F)   
## X 1 11627.5 11627.5 180.5258 < 2e-16 \*\*\*  
## I(X^2) 1 203.1 203.1 3.1538 0.08109 .   
## Residuals 57 3671.3 64.4   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(model2)

##   
## Call:  
## lm(formula = Y ~ X + I(X^2), data = mmass\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -15.086 -6.154 -1.088 6.220 20.578   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 207.349608 29.225118 7.095 2.21e-09 \*\*\*  
## X -2.964323 1.003031 -2.955 0.00453 \*\*   
## I(X^2) 0.014840 0.008357 1.776 0.08109 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 8.026 on 57 degrees of freedom  
## Multiple R-squared: 0.7632, Adjusted R-squared: 0.7549   
## F-statistic: 91.84 on 2 and 57 DF, p-value: < 2.2e-16

### f) Test significance of quadratic term

# p-value for the quadratic term from summary output  
summary(model2)$coefficients["I(X^2)", "Pr(>|t|)"]

## [1] 0.0810869

### g) Third-order model

# Fit third-order model  
model3 <- lm(Y ~ X + I(X^2) + I(X^3), data=mmass\_data)  
summary(model3)

##   
## Call:  
## lm(formula = Y ~ X + I(X^2) + I(X^3), data = mmass\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -15.3671 -5.8483 -0.6755 6.1376 20.0637   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 1.404e+02 1.877e+02 0.748 0.458  
## X 5.648e-01 9.822e+00 0.058 0.954  
## I(X^2) -4.559e-02 1.675e-01 -0.272 0.786  
## I(X^3) 3.369e-04 9.327e-04 0.361 0.719  
##   
## Residual standard error: 8.087 on 56 degrees of freedom  
## Multiple R-squared: 0.7637, Adjusted R-squared: 0.7511   
## F-statistic: 60.34 on 3 and 56 DF, p-value: < 2.2e-16

# Test significance of cubic term  
summary(model3)$coefficients["I(X^3)", "Pr(>|t|)"]

## [1] 0.7192848

## Question 3: CDI Data Set

# Load the CDI data with explicit column classes  
# For large files, it's better to let R determine the classes  
# Try to read the file with fill=TRUE, which will add NAs for missing values  
cdi\_data <- read.table("cdi.txt", header=TRUE, fill=TRUE)  
  
# Check how many rows were read  
nrow(cdi\_data)

## [1] 440

# Examine the structure  
str(cdi\_data)

## 'data.frame': 440 obs. of 17 variables:  
## $ id\_number : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ county : chr "Los\_Angeles" "Cook" "Harris" "San\_Diego" ...  
## $ state : chr "CA" "IL" "TX" "CA" ...  
## $ land\_area\_sq\_mi : int 4060 946 1729 4205 790 71 9204 614 1945 880 ...  
## $ total\_population : int 8863164 5105067 2818199 2498016 2410556 2300664 2122101 2111687 1937094 1852810 ...  
## $ percent\_population\_18\_34 : num 32.1 29.2 31.3 33.5 32.6 28.3 29.2 27.4 27.1 32.6 ...  
## $ percent\_population\_65\_plus : num 9.7 12.4 7.1 10.9 9.2 12.4 12.5 12.5 13.9 8.2 ...  
## $ number\_active\_physicians : int 23677 15153 7553 5905 6062 4861 4320 3823 6274 4718 ...  
## $ number\_hospital\_beds : int 27700 21550 12449 6179 6369 8942 6104 9490 8840 6934 ...  
## $ total\_serious\_crimes : int 688936 436936 253526 173821 144524 680966 177593 193978 244725 214258 ...  
## $ percent\_high\_school\_graduates : num 70 73.4 74.9 81.9 81.2 63.7 81.5 70 65 77.1 ...  
## $ percent\_bachelors\_degrees : num 22.3 22.8 25.4 25.3 27.8 16.6 22.1 13.7 18.8 26.3 ...  
## $ percent\_below\_poverty\_level : num 11.6 11.1 12.5 8.1 5.2 19.5 8.8 16.9 14.2 10.4 ...  
## $ percent\_unemployment : num 8 7.2 5.7 6.1 4.8 9.5 4.9 10 8.7 6.1 ...  
## $ per\_capita\_income : int 20786 21729 19517 19588 24400 16803 18042 17461 17823 21001 ...  
## $ total\_personal\_income\_millions: int 184230 110928 55003 48931 58818 38658 38287 36872 34525 38911 ...  
## $ geographic\_region : int 4 2 3 4 4 1 4 2 3 3 ...

# Look at the last few rows to see if there are issues  
tail(cdi\_data, 10)

## id\_number county state land\_area\_sq\_mi total\_population  
## 431 431 Sarpy NE 241 102583  
## 432 432 Windham CT 513 102525  
## 433 433 Kings CA 1390 101469  
## 434 434 Wayne OH 555 101461  
## 435 435 Charles MD 461 101154  
## 436 436 Hernando FL 478 101115  
## 437 437 Martin FL 556 100900  
## 438 438 Montgomery TN 539 100498  
## 439 439 Maui HI 1159 100374  
## 440 440 Morgan AL 582 100043  
## percent\_population\_18\_34 percent\_population\_65\_plus  
## 431 30.4 4.8  
## 432 28.5 12.5  
## 433 33.7 7.7  
## 434 26.3 11.6  
## 435 29.9 6.5  
## 436 16.4 30.7  
## 437 20.4 27.5  
## 438 35.7 7.9  
## 439 26.2 11.3  
## 440 26.3 11.7  
## number\_active\_physicians number\_hospital\_beds total\_serious\_crimes  
## 431 39 160 2689  
## 432 123 254 1397  
## 433 82 180 4449  
## 434 84 155 2377  
## 435 67 104 5279  
## 436 98 290 4414  
## 437 193 277 5081  
## 438 87 188 6537  
## 439 192 182 7130  
## 440 122 464 4693  
## percent\_high\_school\_graduates percent\_bachelors\_degrees  
## 431 91.0 25.4  
## 432 71.1 16.8  
## 433 65.6 9.0  
## 434 73.6 13.9  
## 435 81.0 16.2  
## 436 70.5 9.7  
## 437 79.7 20.3  
## 438 77.9 16.5  
## 439 77.0 17.8  
## 440 69.4 15.5  
## percent\_below\_poverty\_level percent\_unemployment per\_capita\_income  
## 431 3.5 2.6 16137  
## 432 6.0 9.2 18070  
## 433 15.0 12.8 13907  
## 434 8.4 5.9 16464  
## 435 3.7 4.9 19317  
## 436 7.9 8.2 13919  
## 437 5.0 9.8 27125  
## 438 10.8 8.0 13169  
## 439 5.7 3.2 18504  
## 440 9.4 7.1 16458  
## total\_personal\_income\_millions geographic\_region  
## 431 1655 2  
## 432 1853 1  
## 433 1411 4  
## 434 1670 2  
## 435 1954 3  
## 436 1407 3  
## 437 2737 3  
## 438 1323 3  
## 439 1857 4  
## 440 1647 3

# If there are rows with NA values at the end, you can remove them  
# Remove rows with NA values  
cdi\_data <- na.omit(cdi\_data)  
  
# Check the structure again  
str(cdi\_data)

## 'data.frame': 440 obs. of 17 variables:  
## $ id\_number : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ county : chr "Los\_Angeles" "Cook" "Harris" "San\_Diego" ...  
## $ state : chr "CA" "IL" "TX" "CA" ...  
## $ land\_area\_sq\_mi : int 4060 946 1729 4205 790 71 9204 614 1945 880 ...  
## $ total\_population : int 8863164 5105067 2818199 2498016 2410556 2300664 2122101 2111687 1937094 1852810 ...  
## $ percent\_population\_18\_34 : num 32.1 29.2 31.3 33.5 32.6 28.3 29.2 27.4 27.1 32.6 ...  
## $ percent\_population\_65\_plus : num 9.7 12.4 7.1 10.9 9.2 12.4 12.5 12.5 13.9 8.2 ...  
## $ number\_active\_physicians : int 23677 15153 7553 5905 6062 4861 4320 3823 6274 4718 ...  
## $ number\_hospital\_beds : int 27700 21550 12449 6179 6369 8942 6104 9490 8840 6934 ...  
## $ total\_serious\_crimes : int 688936 436936 253526 173821 144524 680966 177593 193978 244725 214258 ...  
## $ percent\_high\_school\_graduates : num 70 73.4 74.9 81.9 81.2 63.7 81.5 70 65 77.1 ...  
## $ percent\_bachelors\_degrees : num 22.3 22.8 25.4 25.3 27.8 16.6 22.1 13.7 18.8 26.3 ...  
## $ percent\_below\_poverty\_level : num 11.6 11.1 12.5 8.1 5.2 19.5 8.8 16.9 14.2 10.4 ...  
## $ percent\_unemployment : num 8 7.2 5.7 6.1 4.8 9.5 4.9 10 8.7 6.1 ...  
## $ per\_capita\_income : int 20786 21729 19517 19588 24400 16803 18042 17461 17823 21001 ...  
## $ total\_personal\_income\_millions: int 184230 110928 55003 48931 58818 38658 38287 36872 34525 38911 ...  
## $ geographic\_region : int 4 2 3 4 4 1 4 2 3 3 ...

### a) Multiple linear regression with qualitative predictors

# Create dummy variables for the geographic region  
# Geographic region 1 (NE) will be the reference category  
cdi\_data$X3 <- ifelse(cdi\_data$geographic\_region == 2, 1, 0) # North Central  
cdi\_data$X4 <- ifelse(cdi\_data$geographic\_region == 3, 1, 0) # South  
cdi\_data$X5 <- ifelse(cdi\_data$geographic\_region == 4, 1, 0) # West  
  
# Fit the regression model  
cdi\_model <- lm(number\_active\_physicians ~ total\_population + total\_personal\_income\_millions +   
 X3 + X4 + X5, data=cdi\_data)  
summary(cdi\_model)

##   
## Call:  
## lm(formula = number\_active\_physicians ~ total\_population + total\_personal\_income\_millions +   
## X3 + X4 + X5, data = cdi\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1866.8 -207.7 -81.5 72.4 3721.7   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -5.848e+01 5.882e+01 -0.994 0.3207   
## total\_population 5.515e-04 2.835e-04 1.945 0.0524 .   
## total\_personal\_income\_millions 1.070e-01 1.325e-02 8.073 6.8e-15 \*\*\*  
## X3 -3.493e+00 7.881e+01 -0.044 0.9647   
## X4 4.220e+01 7.402e+01 0.570 0.5689   
## X5 -1.490e+02 8.683e+01 -1.716 0.0868 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 566.1 on 434 degrees of freedom  
## Multiple R-squared: 0.9011, Adjusted R-squared: 0.8999   
## F-statistic: 790.7 on 5 and 434 DF, p-value: < 2.2e-16

### b) Interpretation of coefficients

# View coefficients  
coef(cdi\_model)

## (Intercept) total\_population   
## -5.847618e+01 5.514599e-04   
## total\_personal\_income\_millions X3   
## 1.070115e-01 -3.493126e+00   
## X4 X5   
## 4.219673e+01 -1.490196e+02