HW 1

Problem 1

$Y_i = X_i^T B + e_i, \quad i = 1, ---, n$

Prove LSE $\hat{B} = \left(\sum_{i=1}^{n} x_i x_i^T\right)^{-1} \left(\sum_{i=1}^{n} x_i y_i\right) = \underline{(X^T X)^{-1}(X^T Y)}$

$$y = XB + e$$

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & \cdots \\ \vdots & x_{22} & \\ \vdots & & \ddots \\ \vdots & & & x_{Nk} \end{bmatrix} * \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_K \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix}$$

We want to minimize $e^T e$

$$e^T e = \begin{bmatrix} e_1 & e_2 & \cdots & e_N \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix} = \sum_{i=1}^{N} e_i^2$$

$$\min(e^T e) = (Y - XB)^T (Y - XB)$$

$$\min(e^T e) = Y^T Y - 2B^T X^T Y + B^T X^T X B$$

$$\frac{\partial(e^T e)}{\partial B} = -2X^T Y + 2X^T X B$$

set $\frac{\partial(e^T e)}{\partial B} = 0 \Rightarrow 2(-X^T Y + X^T X B) = 0$

$$\therefore X^T Y = X^T X B$$

$$\boxed{\therefore B = (X^T X)^{-1}(X^T Y)}$$

R Notebook

Linear Regression

In R, we must make sure our data meets the assumptions for linear regression. Hence, we check for independence of observations, normality, linearity, and homoscedasticity. In order to perform the linear regression analysis we make use of the lm() function and can interpret the coefficients as well as the p-value - to identify if there is a significant relationship between our dependent and independent variable(s). We can create residual plots with plot() command in R in order to observe if our data meets the assumptions of homoscedasticity. We want the mean of our residuals (i.e. our unexplained variance) to be horizontal and centered around zero. This tells us there are no outliers that could invalidate our linear regression. Additionally, we want the the standardized residuals of Normal Q-Q plot to form a one-to-one line with the theoretical residuals.

Key results for linear regression are seen below:

```
library(ggplot2)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.2

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(broom)
library(ggpubr)
```

```
income.data = read.csv("income.data.csv")
summary(income.data)
```

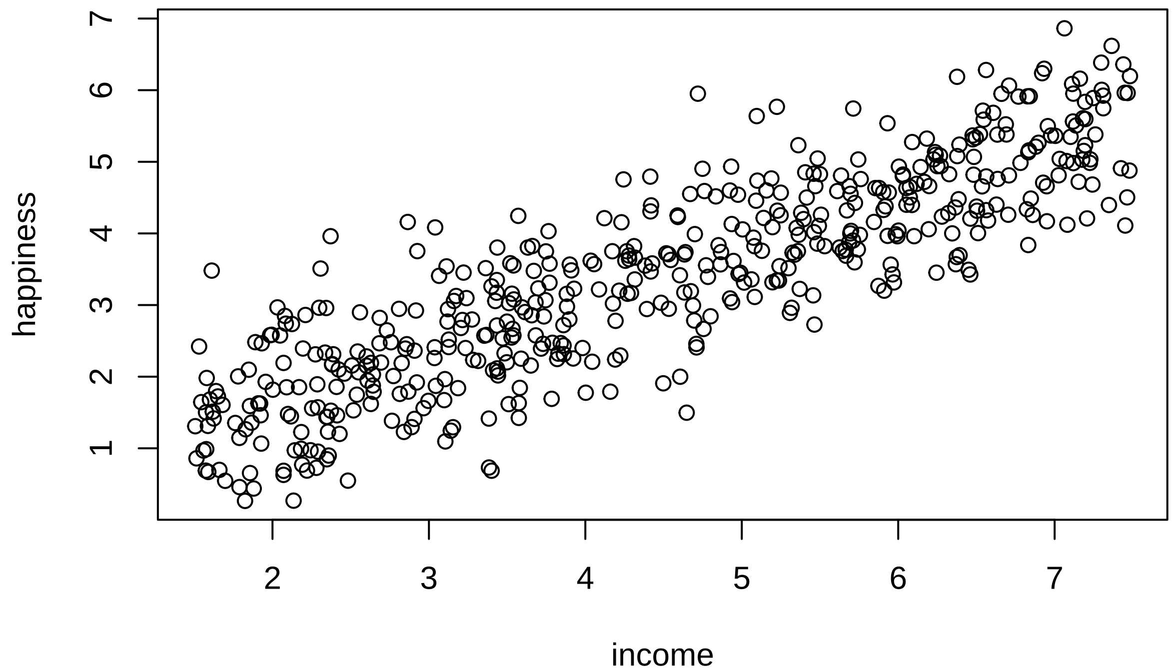
```
##        X              income        happiness
##  Min.   :  1.0   Min.   :1.506   Min.   :0.266
##  1st Qu.:125.2   1st Qu.:3.006   1st Qu.:2.266
##  Median :249.5   Median :4.424   Median :3.473
##  Mean   :249.5   Mean   :4.467   Mean   :3.393
##  3rd Qu.:373.8   3rd Qu.:5.992   3rd Qu.:4.503
##  Max.   :498.0   Max.   :7.482   Max.   :6.863
```

```r
heart.data = read.csv("heart.data.csv")
summary(heart.data)
```

```
##        X             biking          smoking         heart.disease
##  Min.   :  1.0   Min.   : 1.119   Min.   : 0.5259   Min.   : 0.5519
##  1st Qu.:125.2   1st Qu.:20.205   1st Qu.: 8.2798   1st Qu.: 6.5137
##  Median :249.5   Median :35.824   Median :15.8146   Median :10.3853
##  Mean   :249.5   Mean   :37.788   Mean   :15.4350   Mean   :10.1745
##  3rd Qu.:373.8   3rd Qu.:57.853   3rd Qu.:22.5689   3rd Qu.:13.7240
##  Max.   :498.0   Max.   :74.907   Max.   :29.9467   Max.   :20.4535
```

```r
hist(income.data$happiness)
```

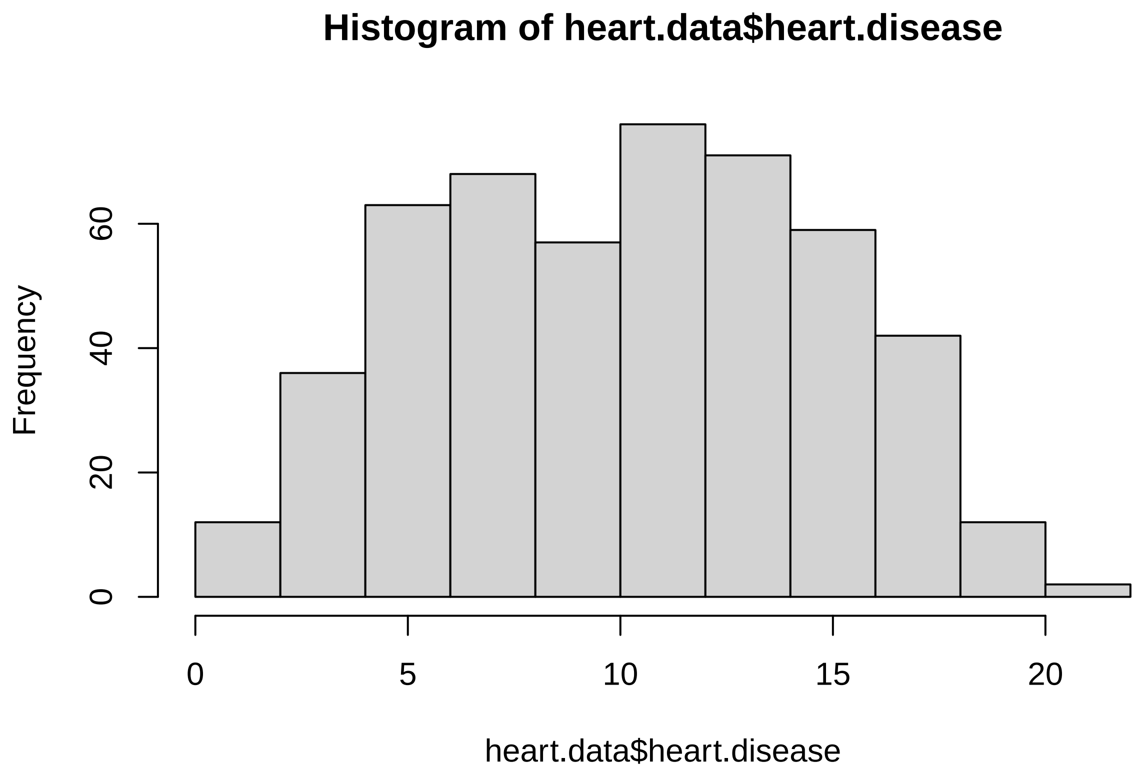

```r
plot(happiness ~ income, data = income.data)
```



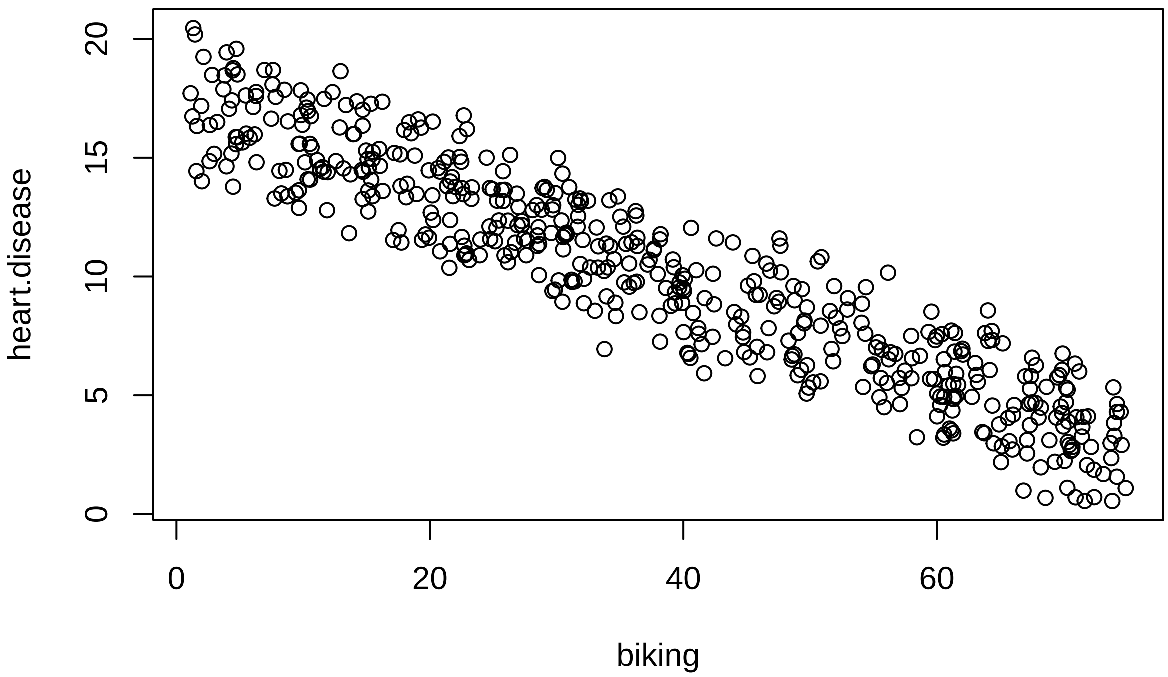

```
cor(heart.data$biking, heart.data$smoking)
```

```
## [1] 0.01513618
```

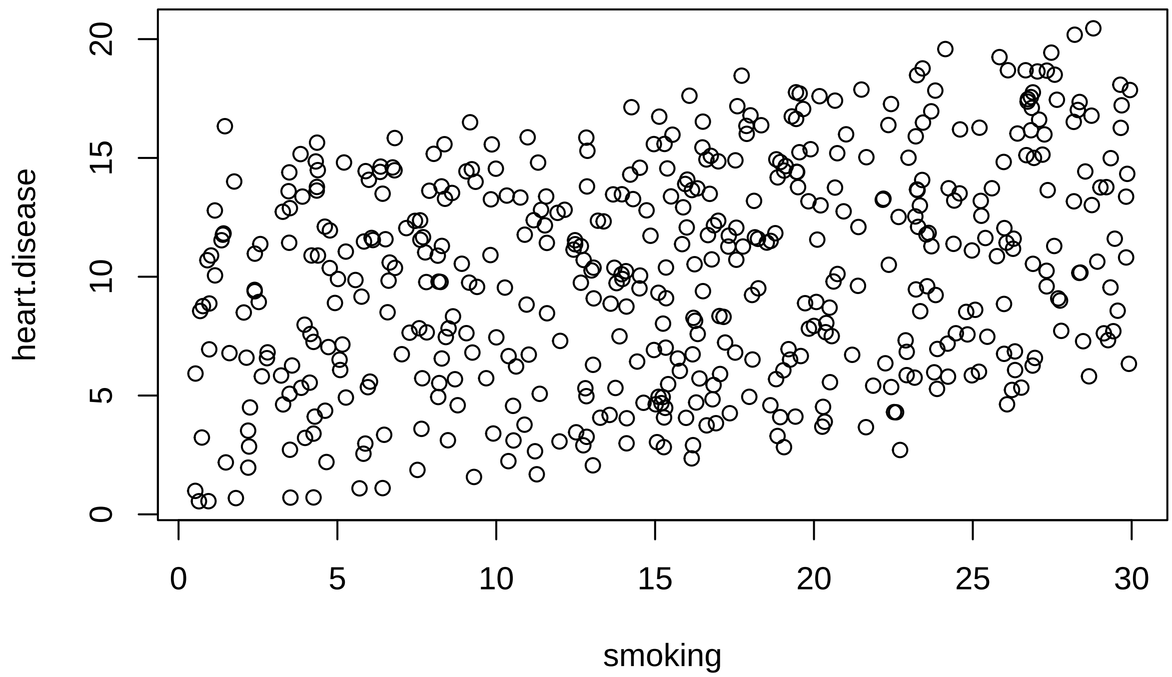
```
hist(heart.data$heart.disease)
```




```
plot(heart.disease ~ biking, data=heart.data)
```




```
plot(heart.disease ~ smoking, data=heart.data)
```




```r
income.happiness.lm <- lm(happiness ~ income, data = income.data)

summary(income.happiness.lm)
```

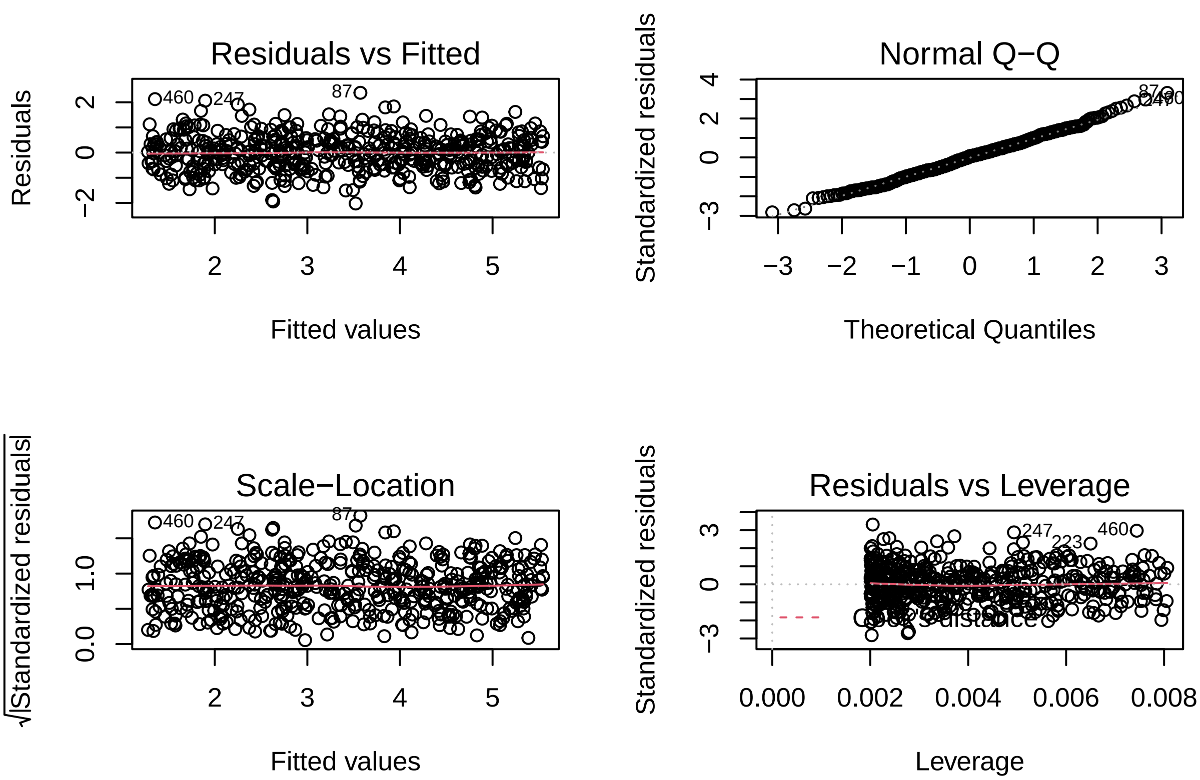
```
##
## Call:
## lm(formula = happiness ~ income, data = income.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.02479 -0.48526  0.04078  0.45898  2.37805
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.20427    0.08884   2.299   0.0219 *
## income       0.71383    0.01854  38.505   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7181 on 496 degrees of freedom
## Multiple R-squared:  0.7493, Adjusted R-squared:  0.7488
## F-statistic:  1483 on 1 and 496 DF,  p-value: < 2.2e-16
```

```r
heart.disease.lm<-lm(heart.disease ~ biking + smoking, data = heart.data)
```

```
summary(heart.disease.lm)
```

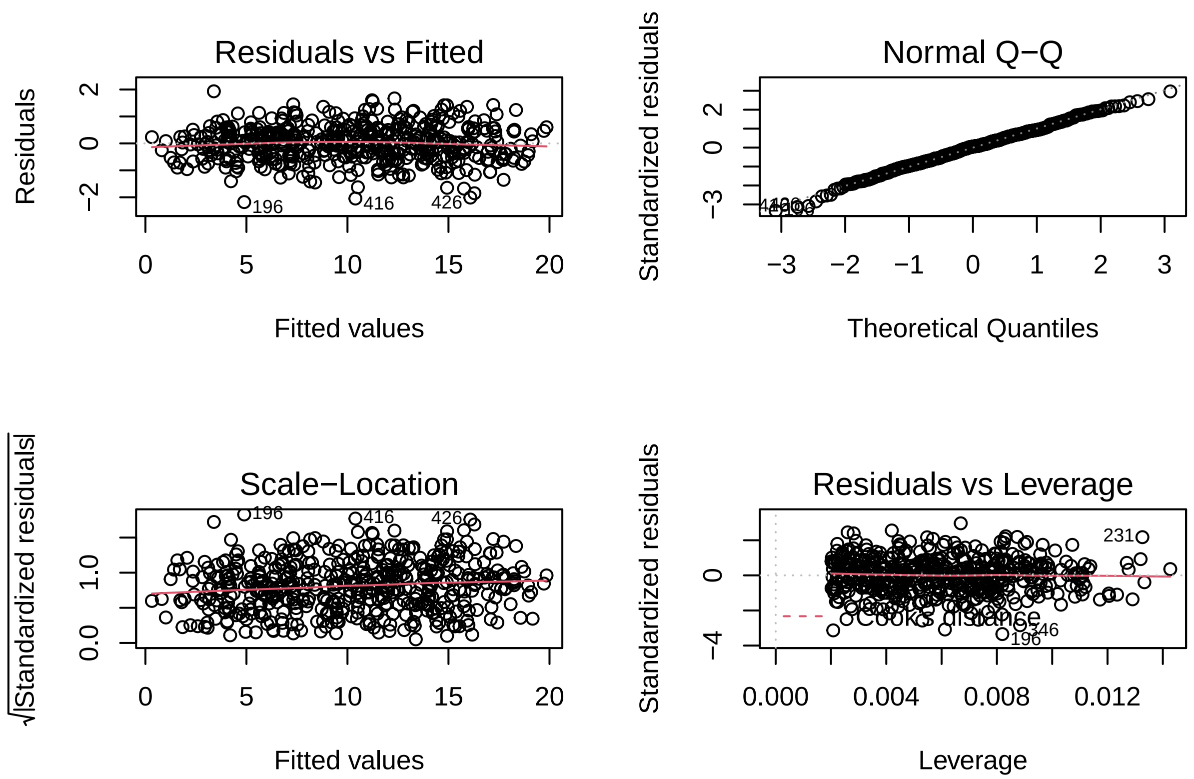
```
##
## Call:
## lm(formula = heart.disease ~ biking + smoking, data = heart.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1789 -0.4463  0.0362  0.4422  1.9331
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 14.984658   0.080137  186.99   <2e-16 ***
## biking      -0.200133   0.001366 -146.53   <2e-16 ***
## smoking      0.178334   0.003539   50.39   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.654 on 495 degrees of freedom
## Multiple R-squared:  0.9796, Adjusted R-squared:  0.9795
## F-statistic: 1.19e+04 on 2 and 495 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(income.happiness.lm)
```



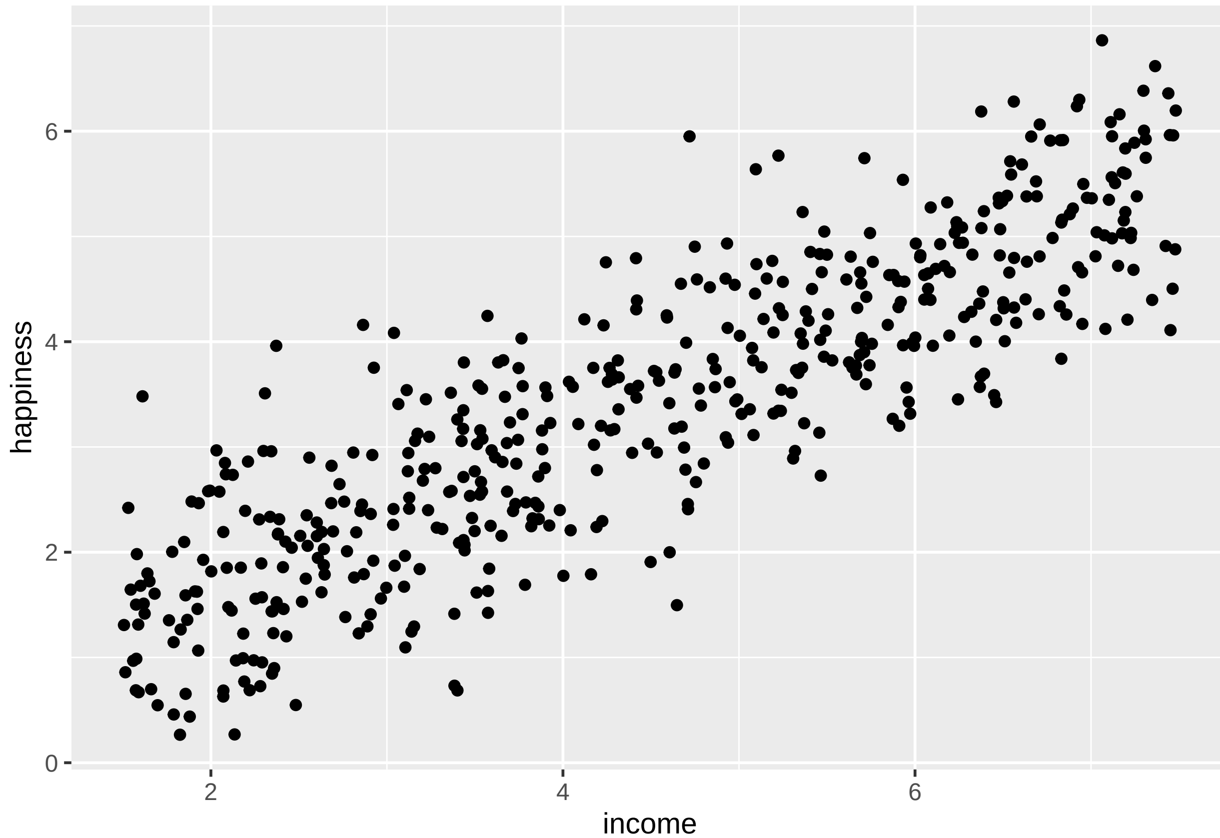
```
par(mfrow=c(1,1))
```

```
par(mfrow=c(2,2))
plot(heart.disease.lm)
```



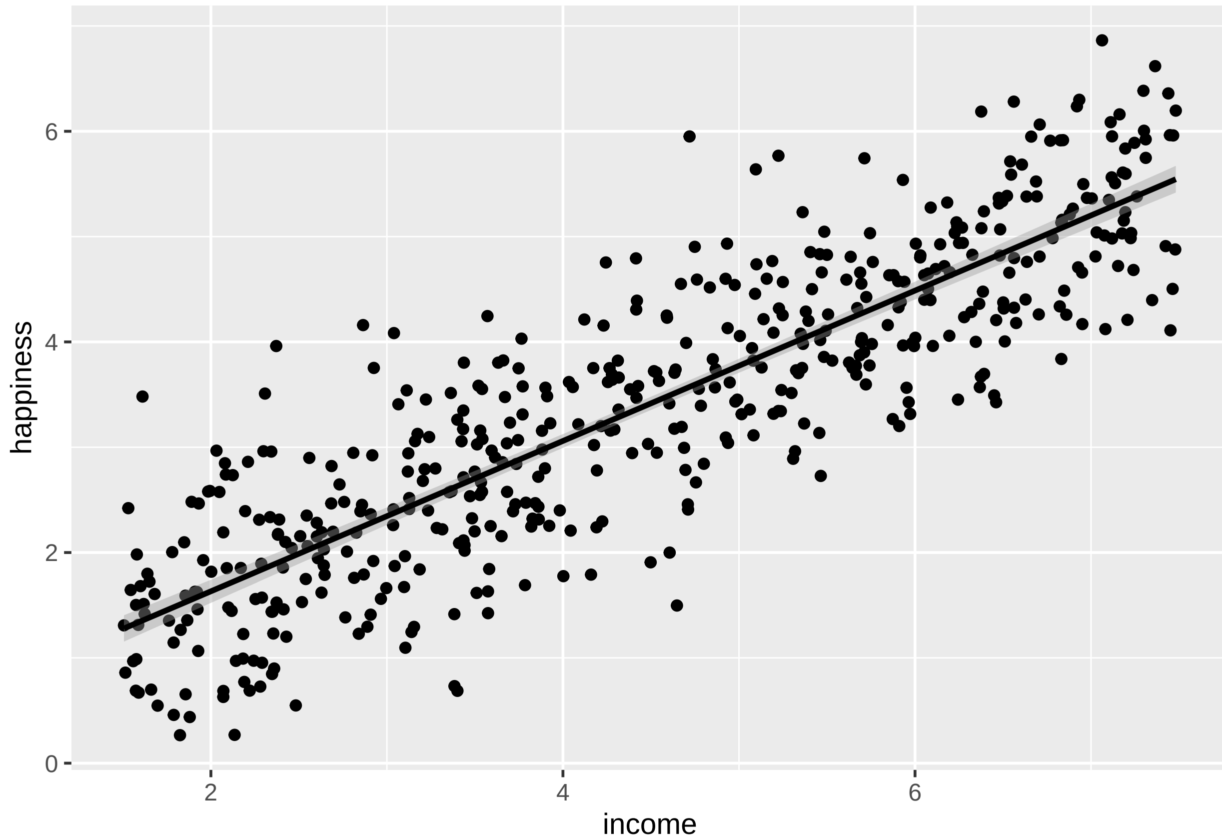

```
par(mfrow=c(1,1))
```

```
income.graph<-ggplot(income.data, aes(x=income, y=happiness))+
                     geom_point()
income.graph
```



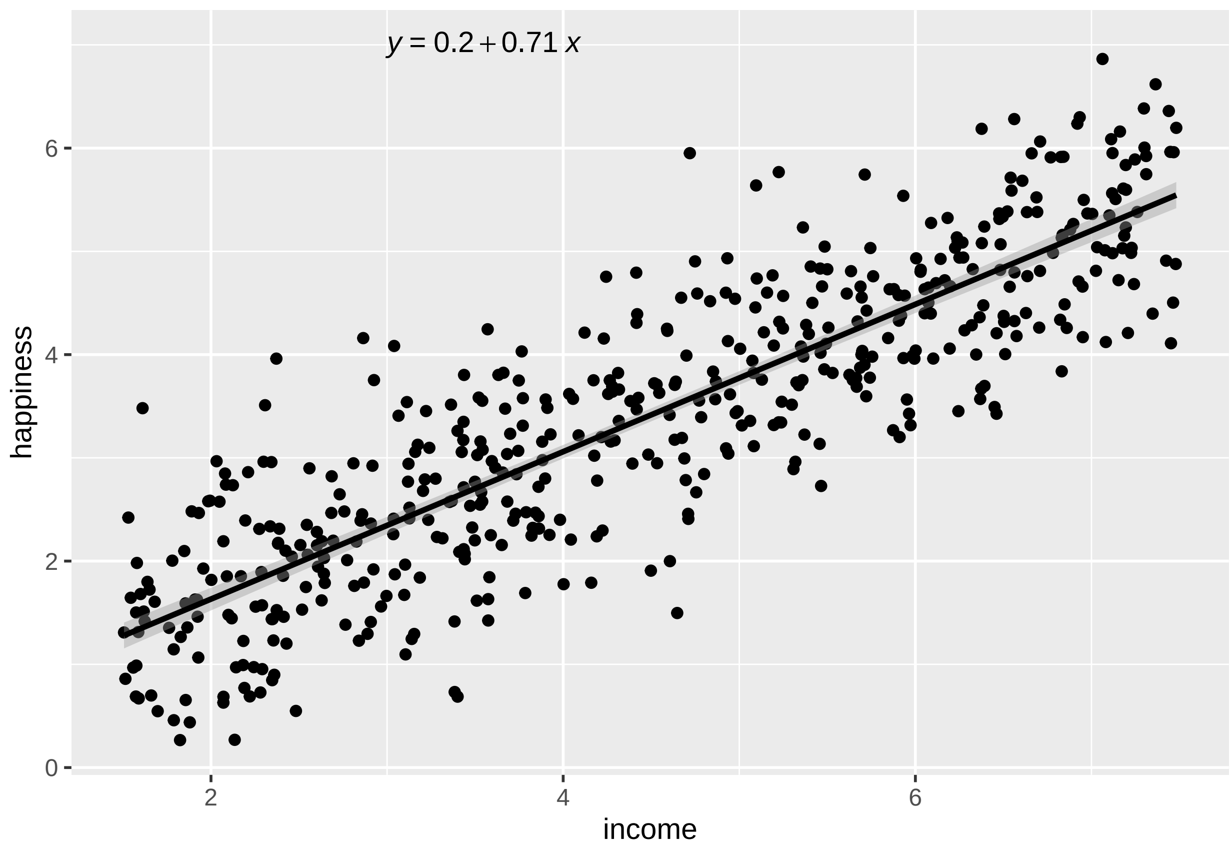

```
income.graph <- income.graph + geom_smooth(method="lm", col="black")

income.graph
```

```
## `geom_smooth()` using formula 'y ~ x'
```



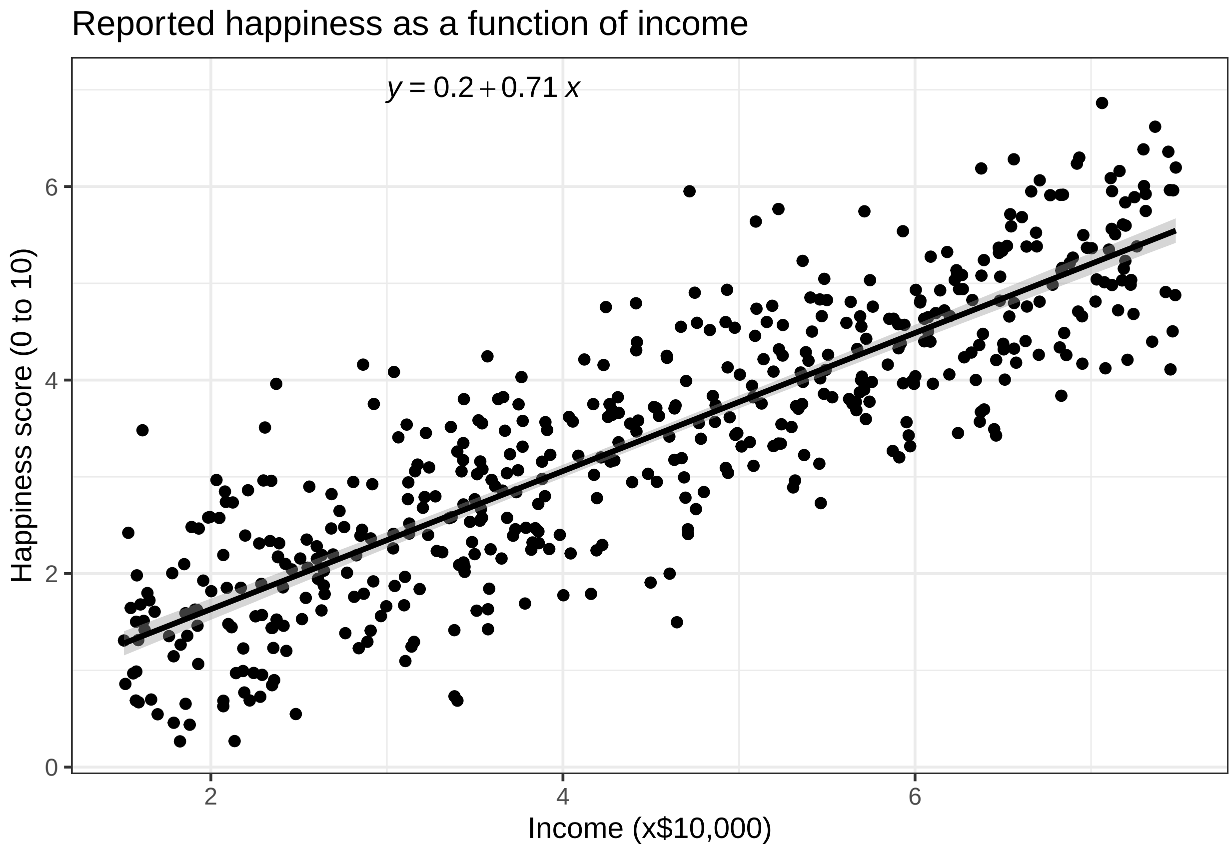

```
income.graph <- income.graph +
  stat_regline_equation(label.x = 3, label.y = 7)

income.graph
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
income.graph +
  theme_bw() +
  labs(title = "Reported happiness as a function of income",
       x = "Income (x$10,000)",
       y = "Happiness score (0 to 10)")
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



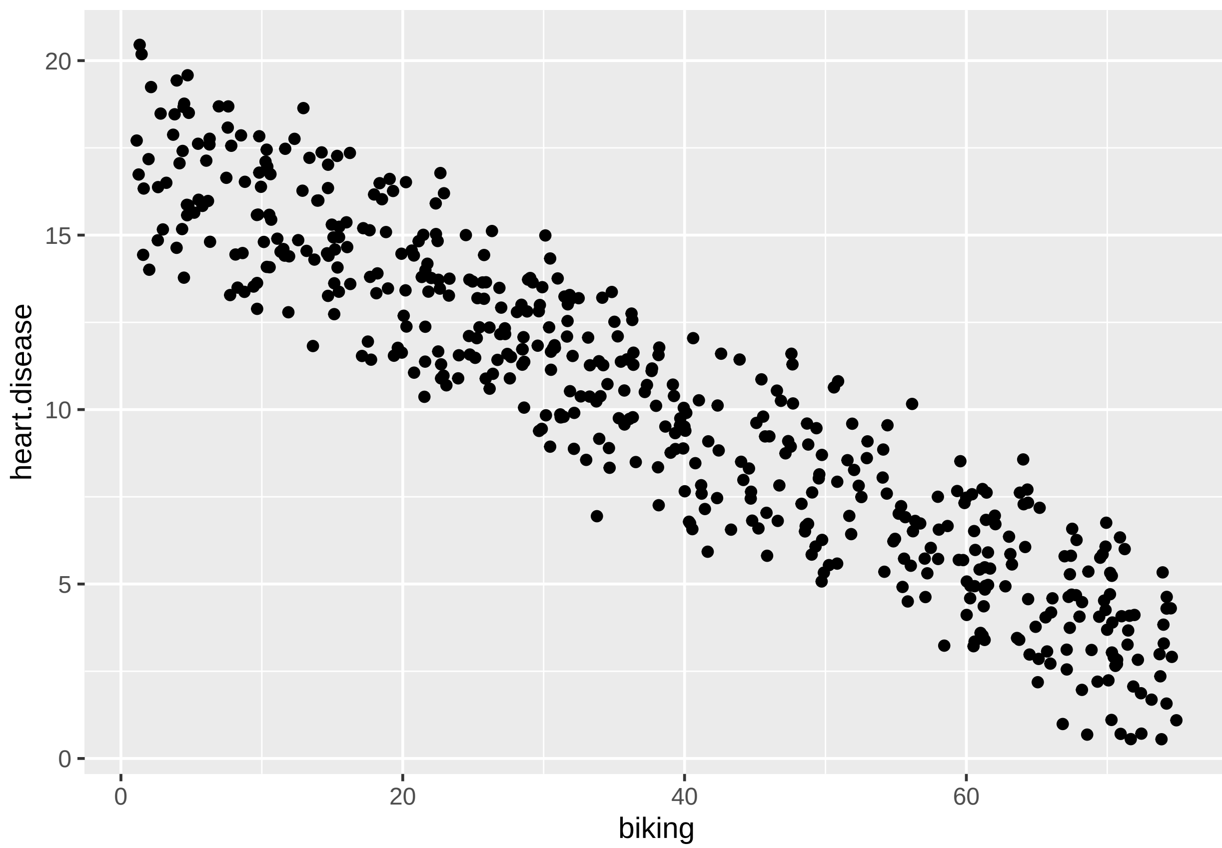

```
plotting.data<-expand.grid(
  biking = seq(min(heart.data$biking), max(heart.data$biking), length.out=30),
    smoking=c(min(heart.data$smoking), mean(heart.data$smoking), max(heart.data$smoking)))
```

```
plotting.data$predicted.y <- predict.lm(heart.disease.lm, newdata=plotting.data)
```

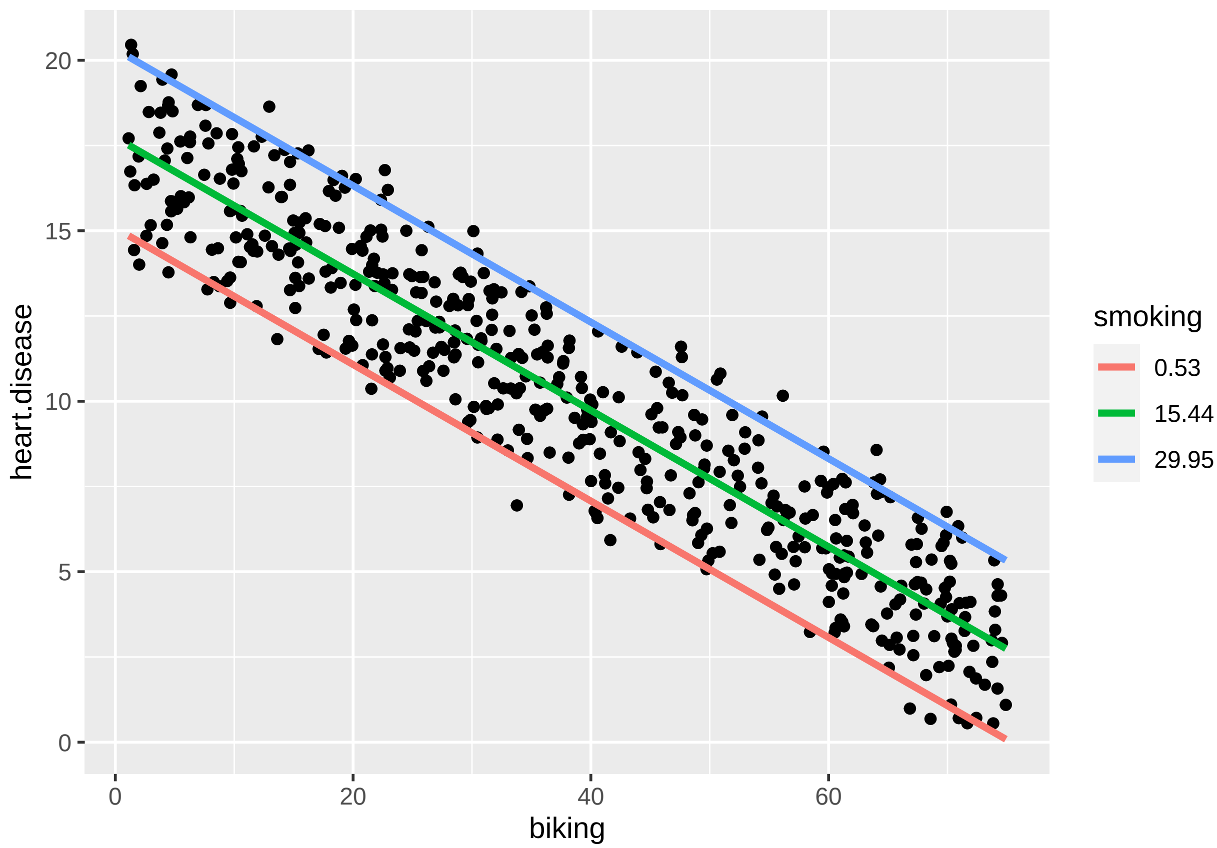
```
plotting.data$smoking <- round(plotting.data$smoking, digits = 2)
```

```
plotting.data$smoking <- as.factor(plotting.data$smoking)
```

```
heart.plot <- ggplot(heart.data, aes(x=biking, y=heart.disease)) +
  geom_point()

heart.plot
```

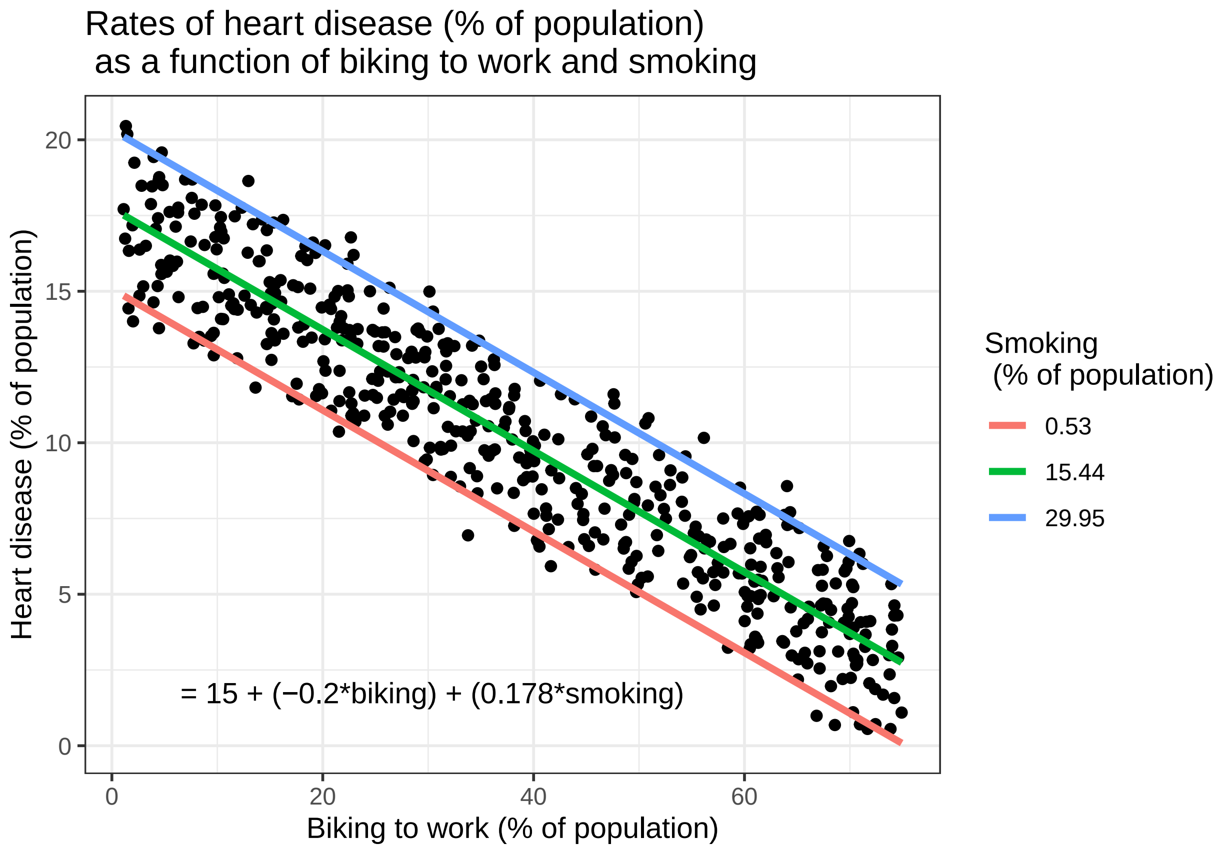



```
heart.plot <- heart.plot +
  geom_line(data=plotting.data, aes(x=biking, y=predicted.y, color=smoking), size=1.25)

heart.plot
```




```
heart.plot <-
heart.plot +
  theme_bw() +
  labs(title = "Rates of heart disease (% of population) \n as a function of biking to work and smoking
      x = "Biking to work (% of population)",
      y = "Heart disease (% of population)",
      color = "Smoking \n (% of population)")

heart.plot
```


```
heart.plot + annotate(geom="text", x=30, y=1.75, label=" = 15 + (-0.2*biking) + (0.178*smoking)")
```



Logistic Regression

When performing logistic regression in R we are interested in a binary outcome. As a result, often we must transform our dependent variable (i.e. our response variable) into a binary form. In order to fit a model with logistic regression we use the glm() function in R. If we observe significant p-values for the coefficients of our predictors we can find which predictors have a significant contribution to our response variable. In doing so we can "re-fit" our logistic regression model by including only significant variables. Additionally, in order to compare different models we can perform an anova to see if a "reduced model" outperformed/underperformed a "full model." When interpreting our model parameters we can see if the odds of our independent variable increase/decrease with different predictors. In order to gauge the impact of each of our predictors towards our response variable we take the exponent of our model coefficients. The reason we do this is logistic regression models the response variable to log(odds). In other words our model coefficients represent a change in log(odds) in our response variable for a unit change in our predictor variable. In order to predict the outcome of our model on unseen data we can use the predict() function in R. Lastly, we can evaluate overdispersion. Overdispersion happens when data has more variability than expected under a given distribution. In logistic regression we check the expected variance for data from a binomial distribution. In order to check for overdispersion we can fit a logistic regression model with two separate distributions (binomial and quasibinomial). If there is a statistically significant differences in the expected variance then overdispersion is a problem in our model.

Key results for logistic regression are seen below:

```
#install.packages("AER")
library("AER")
```

```
## Warning: package 'AER' was built under R version 4.1.3
```

```
## Loading required package: car

## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

## Loading required package: lmtest

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

## Warning: package 'sandwich' was built under R version 4.1.2

## Loading required package: survival
```

```
data(Affairs, package="AER")
View(Affairs)
#summary(Affairs)
```

```
table(Affairs$affairs)
```

```
##
##   0   1   2   3   7  12
## 451  34  17  19  42  38
```

```
Affairs$ynaffair[Affairs$affairs >  0] <- 1
Affairs$ynaffair[Affairs$affairs == 0] <- 0
Affairs$ynaffair <- factor(Affairs$ynaffair,levels=c(0,1), labels=c("No","Yes"))
table(Affairs$ynaffair)
```

```
##
##  No Yes
## 451 150
```

```r
fit.full <- glm(ynaffair ~ gender + age + yearsmarried + children
             + religiousness + education + occupation +rating,
             data=Affairs, family=binomial())
summary(fit.full)
```

```
##
## Call:
## glm(formula = ynaffair ~ gender + age + yearsmarried + children +
##     religiousness + education + occupation + rating, family = binomial(),
##     data = Affairs)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5713  -0.7499  -0.5690  -0.2539   2.5191
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.37726    0.88776   1.551 0.120807
## gendermale     0.28029    0.23909   1.172 0.241083
## age           -0.04426    0.01825  -2.425 0.015301 *
## yearsmarried   0.09477    0.03221   2.942 0.003262 **
## childrenyes    0.39767    0.29151   1.364 0.172508
## religiousness -0.32472    0.08975  -3.618 0.000297 ***
## education      0.02105    0.05051   0.417 0.676851
## occupation     0.03092    0.07178   0.431 0.666630
## rating        -0.46845    0.09091  -5.153 2.56e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 675.38  on 600  degrees of freedom
## Residual deviance: 609.51  on 592  degrees of freedom
## AIC: 627.51
##
## Number of Fisher Scoring iterations: 4
```

```r
fit.reduced   <- glm(ynaffair ~ age + yearsmarried + religiousness
                  + rating, data=Affairs, family=binomial())
summary(fit.reduced)
```

```
##
## Call:
## glm(formula = ynaffair ~ age + yearsmarried + religiousness +
##     rating, family = binomial(), data = Affairs)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6278  -0.7550  -0.5701  -0.2624   2.3998
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)    1.93083    0.61032   3.164 0.001558 **
## age           -0.03527    0.01736  -2.032 0.042127 *
## yearsmarried   0.10062    0.02921   3.445 0.000571 ***
## religiousness -0.32902    0.08945  -3.678 0.000235 ***
## rating        -0.46136    0.08884  -5.193 2.06e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 675.38  on 600  degrees of freedom
## Residual deviance: 615.36  on 596  degrees of freedom
## AIC: 625.36
##
## Number of Fisher Scoring iterations: 4
```

```
anova(fit.reduced, fit.full, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: ynaffair ~ age + yearsmarried + religiousness + rating
## Model 2: ynaffair ~ gender + age + yearsmarried + children + religiousness +
##     education + occupation + rating
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       596     615.36
## 2       592     609.51  4   5.8474   0.2108
```

```
coef(fit.reduced)
```

```
##   (Intercept)           age  yearsmarried religiousness        rating
##    1.93083017   -0.03527112    0.10062274   -0.32902386   -0.46136144
```

```
exp(coef(fit.reduced))
```

```
##   (Intercept)           age  yearsmarried religiousness        rating
##     6.8952321     0.9653437     1.1058594     0.7196258     0.6304248
```

```
newdata1 <- data.frame(rating=c(1,2,3,4,5),age=mean(Affairs$age),
                        yearsmarried=mean(Affairs$yearsmarried),
                        religiousness=mean(Affairs$religiousness))
newdata1
```

```
##   rating      age yearsmarried religiousness
## 1      1 32.48752     8.177696      3.116473
## 2      2 32.48752     8.177696      3.116473
## 3      3 32.48752     8.177696      3.116473
## 4      4 32.48752     8.177696      3.116473
## 5      5 32.48752     8.177696      3.116473
```

```r
newdata1$prob <- predict(fit.reduced, newdata=newdata1,
                         type="response")
newdata1
```

```
##   rating      age yearsmarried religiousness      prob
## 1      1 32.48752     8.177696      3.116473 0.5302296
## 2      2 32.48752     8.177696      3.116473 0.4157377
## 3      3 32.48752     8.177696      3.116473 0.3096712
## 4      4 32.48752     8.177696      3.116473 0.2204547
## 5      5 32.48752     8.177696      3.116473 0.1513079
```

```r
newdata2 <- data.frame(rating=mean(Affairs$rating), age=c(17,27,37,47,57), yearsmarried=mean(Affairs$yea
                       religiousness=mean(Affairs$religiousness))
newdata2
```

```
##    rating age yearsmarried religiousness
## 1 3.93178  17     8.177696      3.116473
## 2 3.93178  27     8.177696      3.116473
## 3 3.93178  37     8.177696      3.116473
## 4 3.93178  47     8.177696      3.116473
## 5 3.93178  57     8.177696      3.116473
```

```r
newdata2$prob <- predict(fit.reduced, newdata=newdata2,
                         type="response")
newdata2
```

```
##    rating age yearsmarried religiousness      prob
## 1 3.93178  17     8.177696      3.116473 0.3350834
## 2 3.93178  27     8.177696      3.116473 0.2615373
## 3 3.93178  37     8.177696      3.116473 0.1992953
## 4 3.93178  47     8.177696      3.116473 0.1488796
## 5 3.93178  57     8.177696      3.116473 0.1094738
```

```r
deviance(fit.reduced)/df.residual(fit.reduced)
```

```
## [1] 1.03248
```

```r
fit    <- glm(ynaffair ~ age + yearsmarried + religiousness +
             rating, family = binomial(), data = Affairs)
fit.od <- glm(ynaffair ~ age + yearsmarried + religiousness +
             rating, family = quasibinomial(), data = Affairs)
pchisq(summary(fit.od)$dispersion * fit$df.residual,
 fit$df.residual, lower = F)
```

```
## [1] 0.340122
```