Problem statement:

Given a dataset of 500 observations of rates of smoking, biking, and heart disease in 500 imaginary small town, run a linear regression to check for correlation between predictive variables of smoking and biking and dependent variable of heart disease.

==========

Dataset characteristics

==========

Heart.data.csv have the following fields:

1. Biking – No of people who use bike to work
2. Smoking – No of people who are smokers
3. heart.disease – No of people with heart diseases

Part 1: prep

1. Install dependencies– only do this the first time.

```
#install.packages("ggplot2")

# install.packages("dplyr")

# install.packages("broom")

#install.packages("ggpubr")
```

2. Load packages.

```
# load packages


library(ggplot2)

library(dplyr)

library(broom)

library(ggpubr)
```

3. Load and examine data.

```
heart.data =read.csv("C:/Users/Lancy/Desktop/heart.data.csv")

View(heart.data)

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

Part 2: check that data meets assumptions

1. Check for independence of observations (aka no autocorrelation)

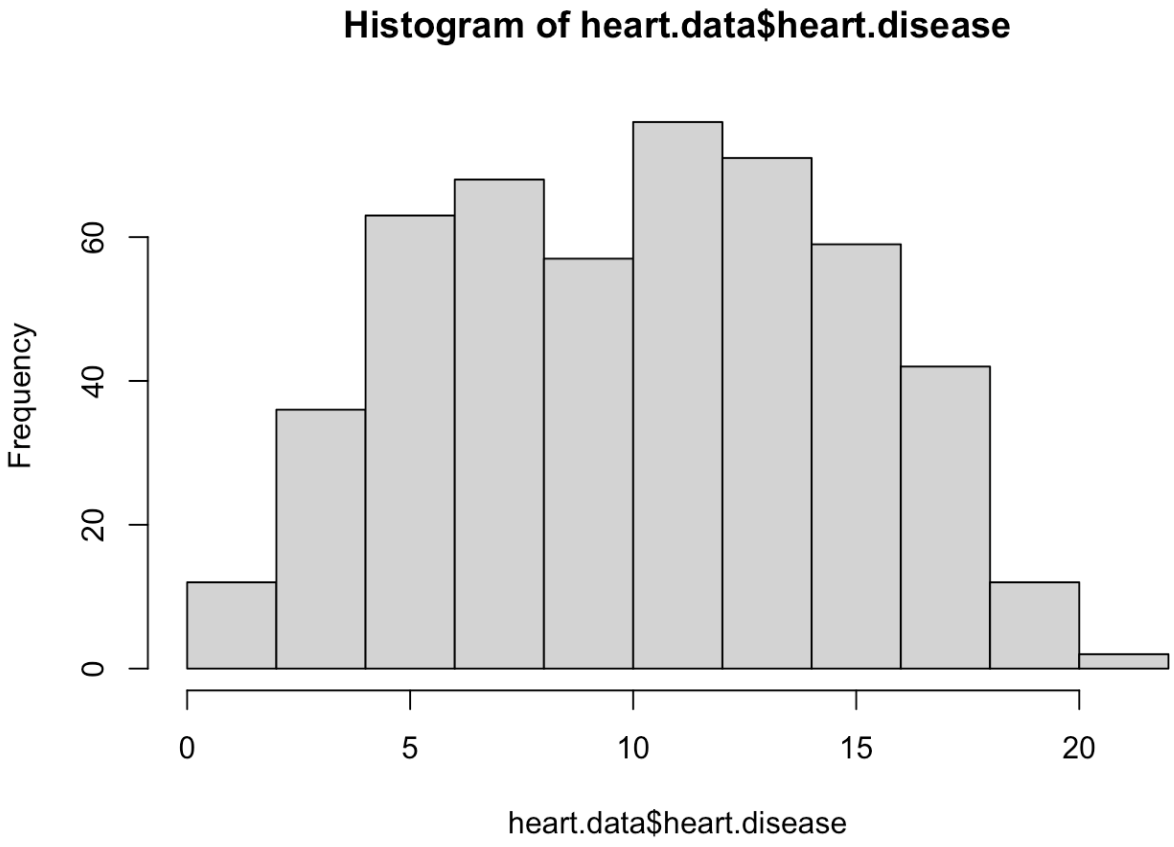
```
cor(heart.data$biking, heart.data$smoking)
```

[1] 0.01513618

The correlation of 1.5% is small enough that we can include both variables in the model.

2. Check for normality of dependent variable (heart disease) with a histogram.

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

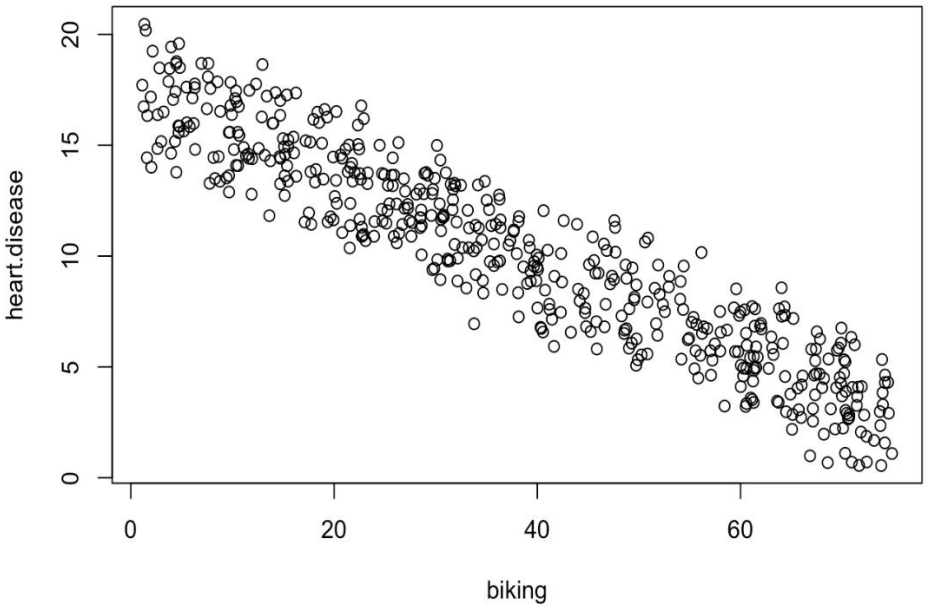


The distribution of observations is roughly bell shaped so we can proceed.

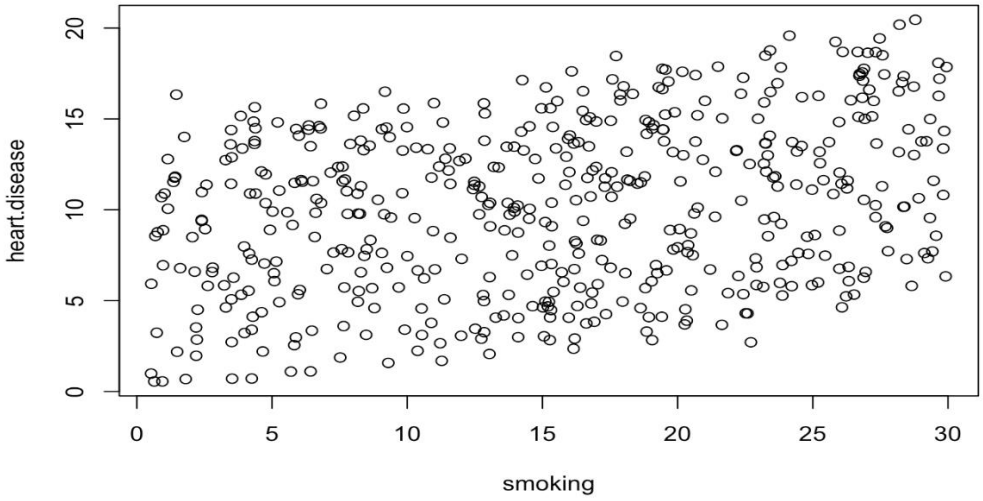
3. Check for linearity.

We'll check this with two scatterplots: for biking and heart disease, and for smoking and heart disease.

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



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



Both look roughly linear

4. Check for homoscedasticity– we'll do this after we make the model.

Part 3: Perform the linear regression.

1. Fit the linear model.

```
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
```

* Estimate for model parameters: value of y-intercept, estimated effect of biking on heart disease, estimated effect of smoking on heart disease. * Standard error of the estimated values * Test statistic aka t value aka t statistic * P value (P >|t|) aka the probability of finding the given t-statistic if the null hypothesis of no relationship were true (we want this to be low!!!)

The final 3 lines are model diagnostics– the most important is the p-value– 2.2e-16 is very close to zero– which indicates that the model fits the data well.

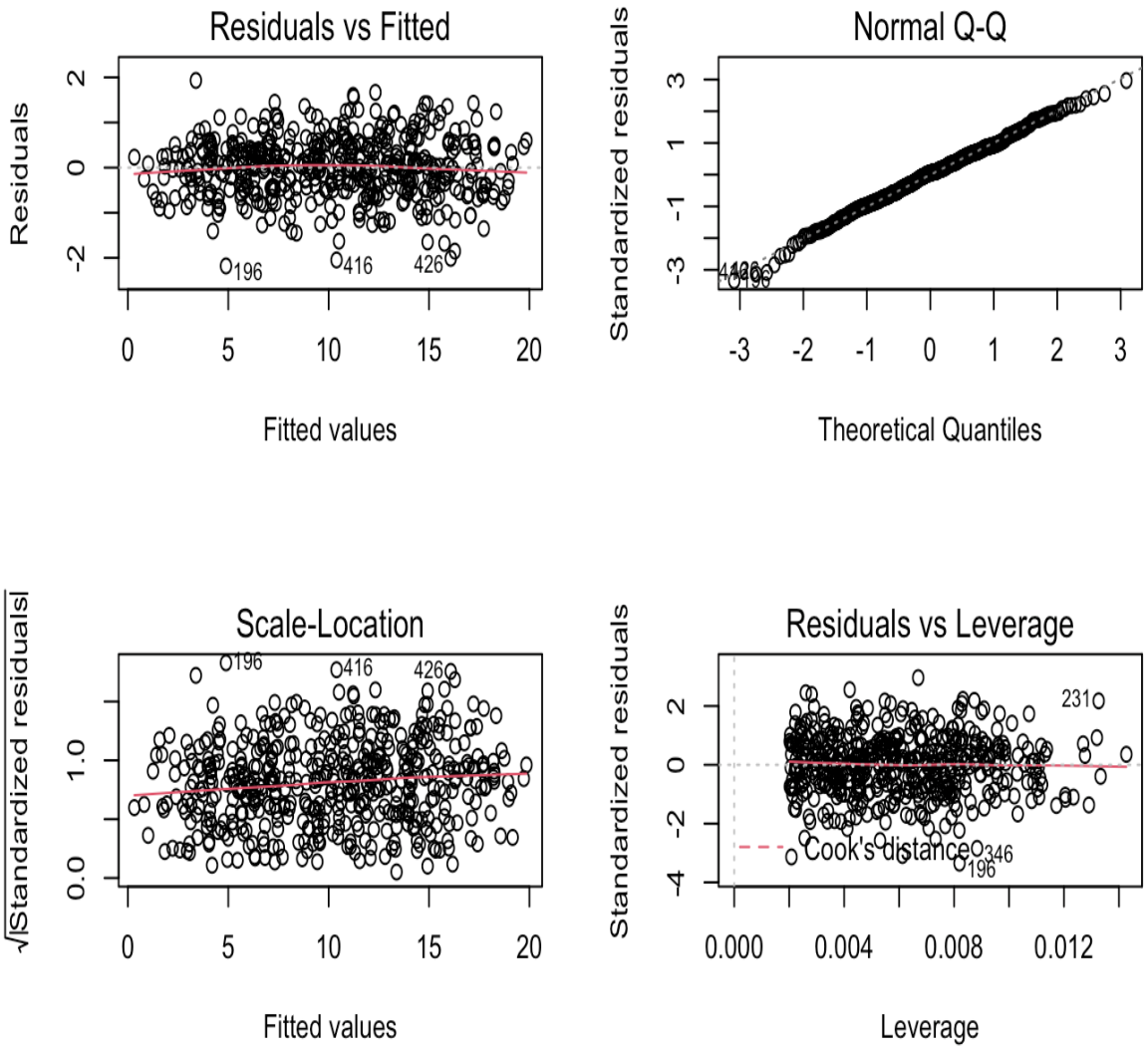
Interpretation: * For every 1% increase in biking to work, there is a correlated 0.2% decrease in heart disease. * For every 1% increase in smoking, there is a correlated 0.178% increase in heart disease.

The standard errors for these regression coefficients are very small, and the t-statistics are very large (-147 and 50.4, respectively). The p-values reflect these small errors and large t-statistics. For both parameters, there is almost zero probability that this effect is due to chance.

Part 4: check for homoscedasticity

par(mfrow=c(2,2))

plot(heart.disease.lm)



par(mfrow=c(1,1))

Residuals are the unexplained variance. They are not exactly the same as model error, but they are calculated from it, so seeing a bias in the residuals would also indicate a bias in the error.

The most important thing to look for is that the red lines representing the mean of the residuals are all basically horizontal and centered around zero. This means there are no outliers or biases in the data that would make a linear regression invalid.

In the Normal Q-Qplot in the top right, we can see that the real residuals from our model form an almost perfectly one-to-one line with the theoretical residuals from a perfect model.

Based on these residuals, we can say that our model meets the assumption of homoscedasticity.

Part 4: Visualize results with a graph.

The visualization step for multiple regression is more difficult than for simple regression, because we now have two predictors. One option is to plot a plane, but these are difficult to read and not often published.

We will try a different method: plotting the relationship between biking and heart disease at different levels of smoking. In this example, smoking will be treated as a factor with three levels, just for the purposes of displaying the relationships in our data.

1. Create a new dataframe with the information you need to plot the model.

```
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))
)
View(plotting.data)
```

2. Predict values of heart disease based on linear model and save them as a new column in our plotting.data

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

3. Round the smoking numbers to two decimals– this will make the legend easier to read.

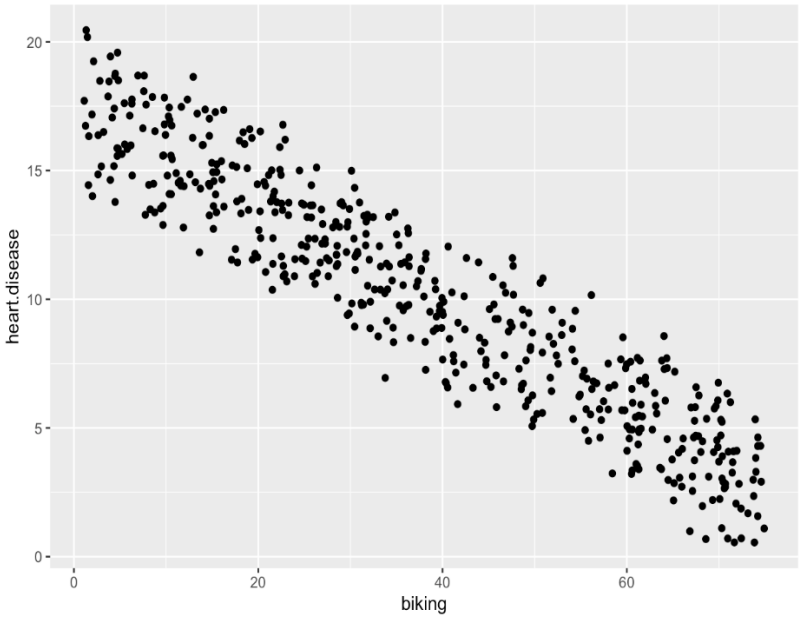
```
plotting.data$smoking <-round(plotting.data$smoking, digits=2)
View(plotting.data)
```

4. Change the smoking variable into a factor so we can plot the interaction between biking and heart dieease at each of the three levels of smoking we chose.

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

5. Plot the original data

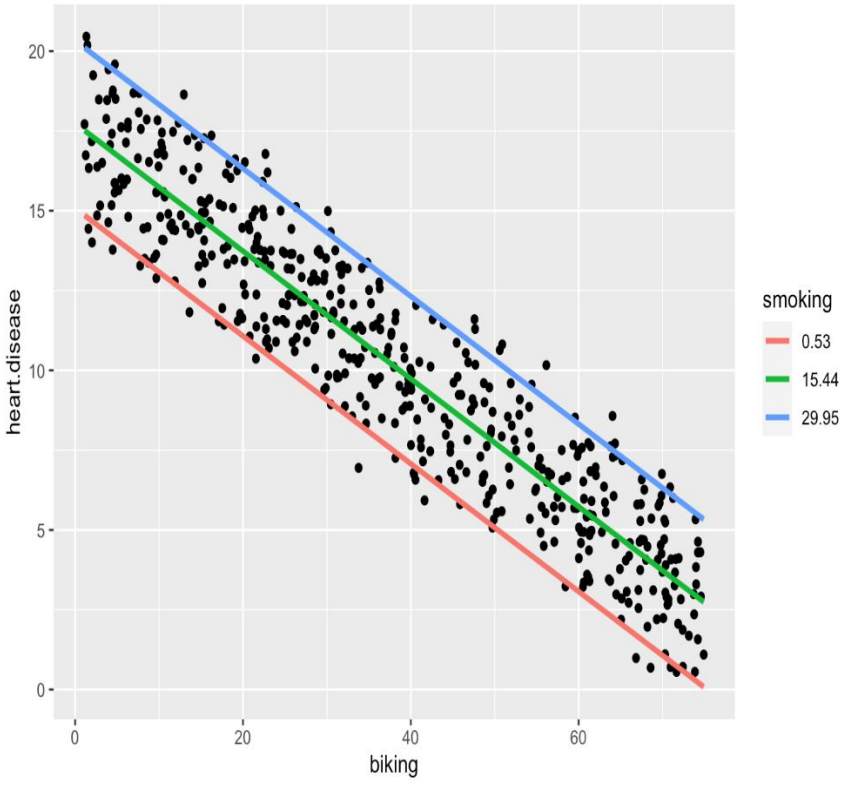
```
heart.plot <- ggplot(heart.data, aes(x=biking, y=heart.disease)) +
  geom_point()


heart.plot
```



6. Add the regression lines

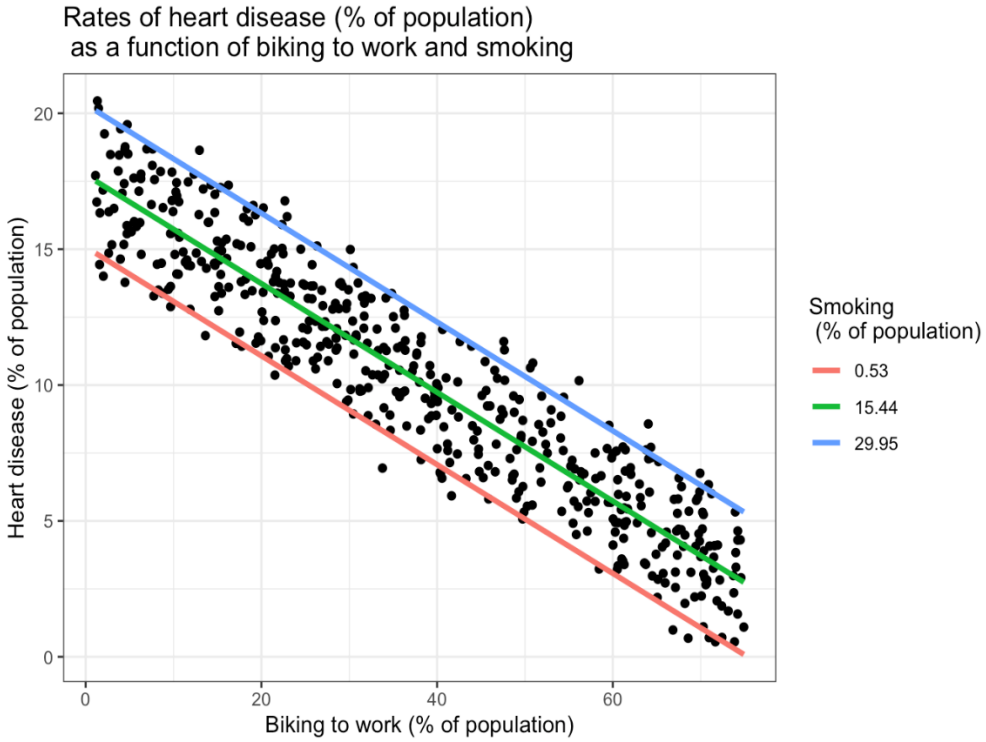
```
heart.plot <- heart.plot +
  geom_line(data=plotting.data, aes(x=biking, y=predicted.y, color=smoking), size=1.25)
```

```
heart.plot
```



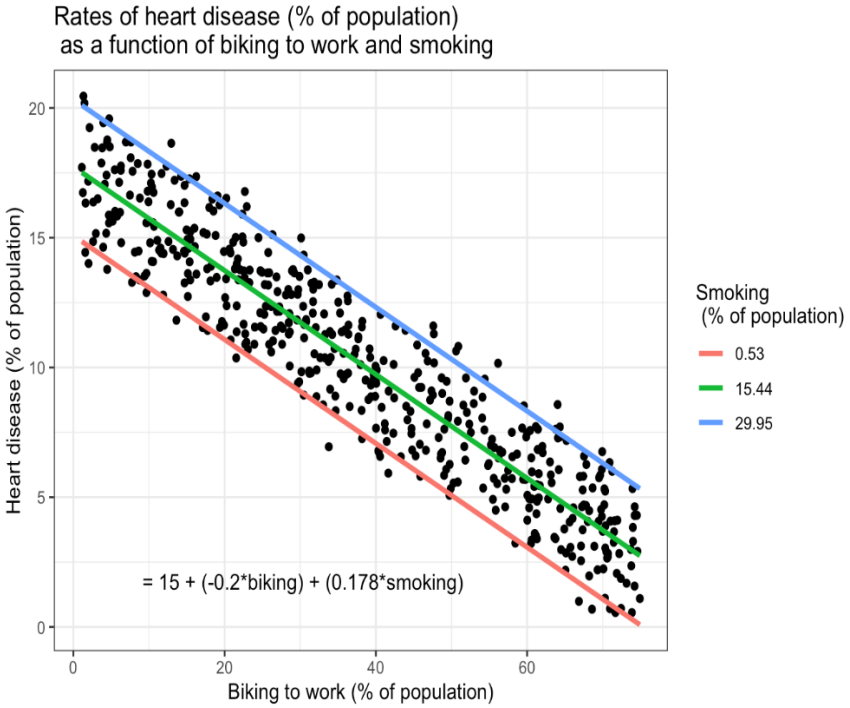
7. Prepare the graph for publication.

```
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
```



8. Add our regression model to the graph.

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



9. Final interpretation when reporting results.

In our survey of 500 towns, we found significant relationships between the frequency of biking to work and the frequency of heart disease and the frequency of smoking and frequency of heart disease ($p < 0$ and $p<0.001$, respectively).

Specifically, we found a 0.2% decrease (± 0.0014) in the frequency of heart disease for every 1% increase in biking, and a 0.178% increase (± 0.0035) in the frequency of heart disease for every 1% increase in smoking.

(Note that we pulled the margin of error from the std error for our coefficients!)