Intro to Data Science HW 7

Copyright Jeffrey Stanton, Jeffrey Saltz, and Jasmina Tacheva

```
# Enter your name here:  Ryan Tervo
# Course Number:         IST 687
# Assignment Name:       Homework #7
# Due Date:              28 Nov 2022
# Submitted Date:        28 Nov 2022
```

Attribution statement: (choose only one and delete the rest)

```
# 1. I did this homework by myself, with help from the book and the professor.
```

The chapter on **linear models** (“Lining Up Our Models”) introduces **linear predictive modeling** using the tool known as **multiple regression**. The term “multiple regression” has an odd history, dating back to an early scientific observation of a phenomenon called **“regression to the mean.”** These days, multiple regression is just an interesting name for using **linear modeling** to assess the **connection between one or more predictor variables and an outcome variable**.

In this exercise, you will **predict Ozone air levels from three predictors**.

A. We will be using the **airquality** data set available in R. Copy it into a dataframe called **air** and use the appropriate functions to **summarize the data**.

```
#   LOAD LIBRARIES:
#library(jsonlite)
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
##    ggplot2 3.4.0     purrr   0.3.5
##    tibble  3.1.8     dplyr   1.0.10
##    tidyr   1.2.1     stringr 1.4.1
##    readr   2.1.3     forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
##    dplyr::filter() masks stats::filter()
##    dplyr::lag()    masks stats::lag()
```

```
library(imputeTS)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
#library(dplyr)
#library(purrr)
#library(maps)
```

```
#   DEFINE VARIABLE:
air <- airquality

#   INITIAL DATA UNDERSTANDING:
str(air)
```

```
## 'data.frame':    153 obs. of  6 variables:
##  $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
##  $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
##  $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
##  $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
##  $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
summary(air)
```

```
##      Ozone           Solar.R           Wind             Temp      
##  Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00  
##  1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00  
##  Median : 31.50   Median :205.0   Median : 9.700   Median :79.00  
##  Mean   : 42.13   Mean   :185.9   Mean   : 9.958   Mean   :77.88  
##  3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00  
##  Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00  
##  NA's   :37       NA's   :7                                       
##      Month            Day      
##  Min.   :5.000   Min.   : 1.0  
##  1st Qu.:6.000   1st Qu.: 8.0  
##  Median :7.000   Median :16.0  
##  Mean   :6.993   Mean   :15.8  
##  3rd Qu.:8.000   3rd Qu.:23.0  
##  Max.   :9.000   Max.   :31.0  
## 
```

```
head(air)
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

B. In the analysis that follows, **Ozone** will be considered as the **outcome variable**, and **Solar.R**, **Wind**, and **Temp** as the **predictors**. Add a comment to briefly explain the outcome and predictor variables in the dataframe using **?airquality**.

```
#   DEFINE THE MODEL:
model <- lm(formula = Ozone ~ Solar.R + Wind + Temp, data = air)
```

```
#   VARIABLE EXPLANATION:
################################################################################
#   ?airquality #  Initially used to inspect airquality dataframe variables.
#   Ozone:    Mean ozone in parts per billion from 1300 to 1500 hours at Roosevelt Island
#   Solar.R:  Solar radiation in Langleys in the frequency band 4000–7700 Angstroms from 0800
to 1200 hours at Central Park
#   Wind:     Average wind speed in miles per hour at 0700 and 1000 hours at LaGuardia Airport
#   Temp:     Maximum daily temperature in degrees Fahrenheit at La Guardia Airport.

##  The goal will be to create a model that can use the predictor variables to determine the o
utcome variable.
```

C. Inspect the outcome and predictor variables – are there any missing values? Show the code you used to check for that.

```
#   INSPECT PREDICTOR VARIABLES:
numNA1 <- sum(is.na(air$Solar.R))
numNA2 <- sum(is.na(air$Wind))
numNA3 <- sum(is.na(air$Temp))
numNA4 <- sum(is.na(air$Ozone))

#   DISPLAY RESULTS:
printString1 <- paste('Missing entries for column Solar.R:  ', numNA1, sep = '')
printString2 <- paste('Missing entries for column Wind:     ', numNA2, sep = '')
printString3 <- paste('Missing entries for column Temp:     ', numNA3, sep = '')
printString4 <- paste('Missing entries for column Ozone:    ', numNA4, sep = '')

print(printString1, quote = FALSE)
```

```
## [1] Missing entries for column Solar.R:  7
```

```
print(printString2, quote = FALSE)
```

```
## [1] Missing entries for column Wind:     0
```

```
print(printString3, quote = FALSE)
```

```
## [1] Missing entries for column Temp:     0
```

```
print(printString4, quote = FALSE)
```

```
## [1] Missing entries for column Ozone:    37
```

D. Use the **na_interpolation()** function from the **imputeTS package** (remember this was used in a previous HW) to fill in the missing values in each of the 4 columns. Make sure there are no more missing values using the commands from Step C.

```
#   LOAD LIBRARY:
```

```
library(imputeTS) #  Did this previously but doing again should not hurt anything.

#   USE 'na_interpolation()' ON 'Solar.R' AND 'Ozone' FIELD TO ELIMINATE MISSING ENTRIES:
air$Solar.R <- na_interpolation(air$Solar.R)
air$Ozone   <- na_interpolation(air$Ozone)

#   VERIFY MISSING FIELDS HAVE BEEN RESOLVED:
numNA1 <- sum(is.na(air$Solar.R))
numNA4 <- sum(is.na(air$Ozone))


printString1 <- paste('Missing entries for column Solar.R:  ', numNA1, sep = '')
print(printString1, quote = FALSE)
```

```
## [1] Missing entries for column Solar.R:  0
```

```
printString4 <- paste('Missing entries for column Ozone:    ', numNA4, sep = '')
print(printString4, quote = FALSE)
```

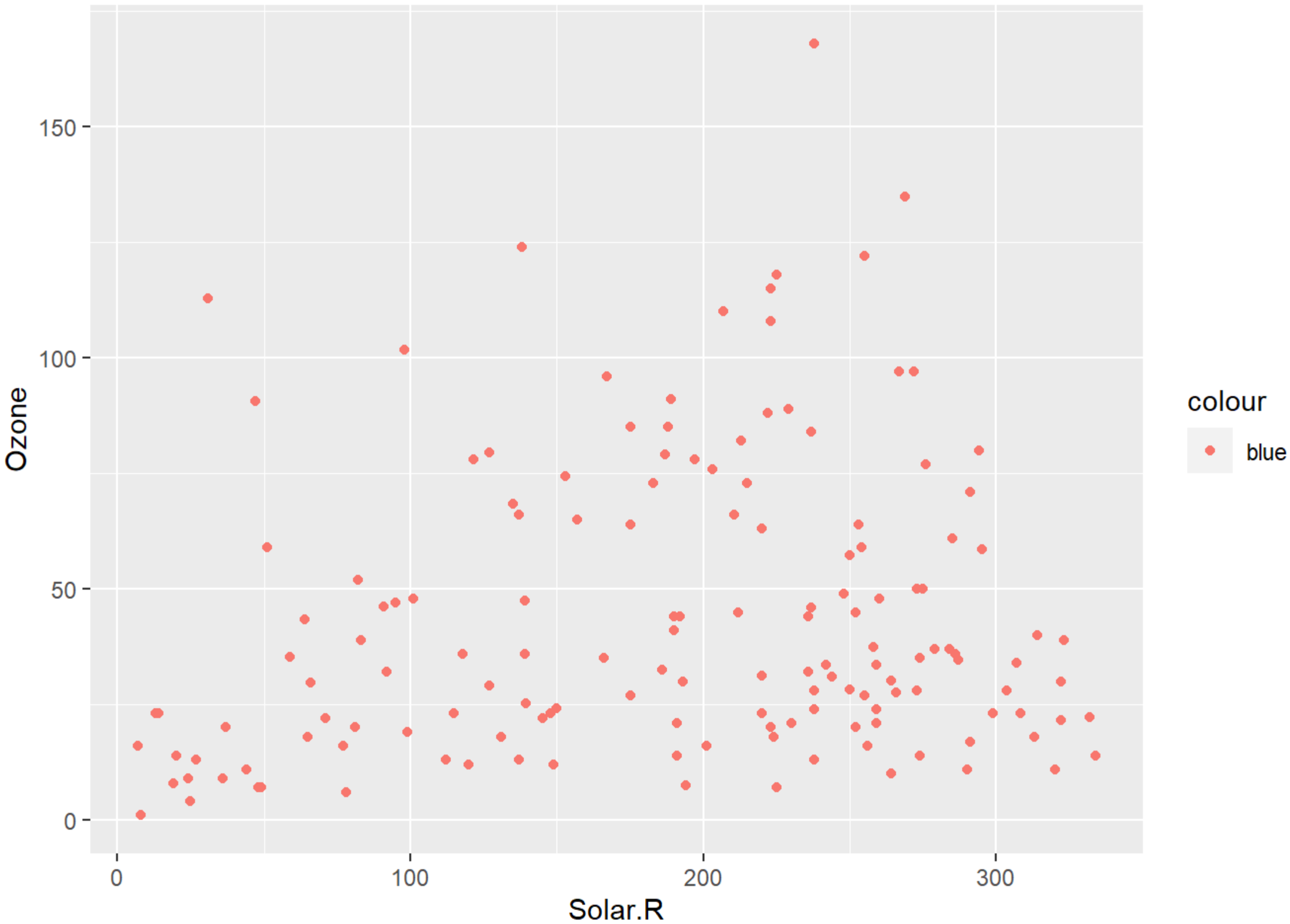
```
## [1] Missing entries for column Ozone:    0
```

```
printString5 <- paste('Other columns did not have any missing data as seen in the prior checki
ng in "Part C".', sep = '')
print(printString5, quote = FALSE)
```

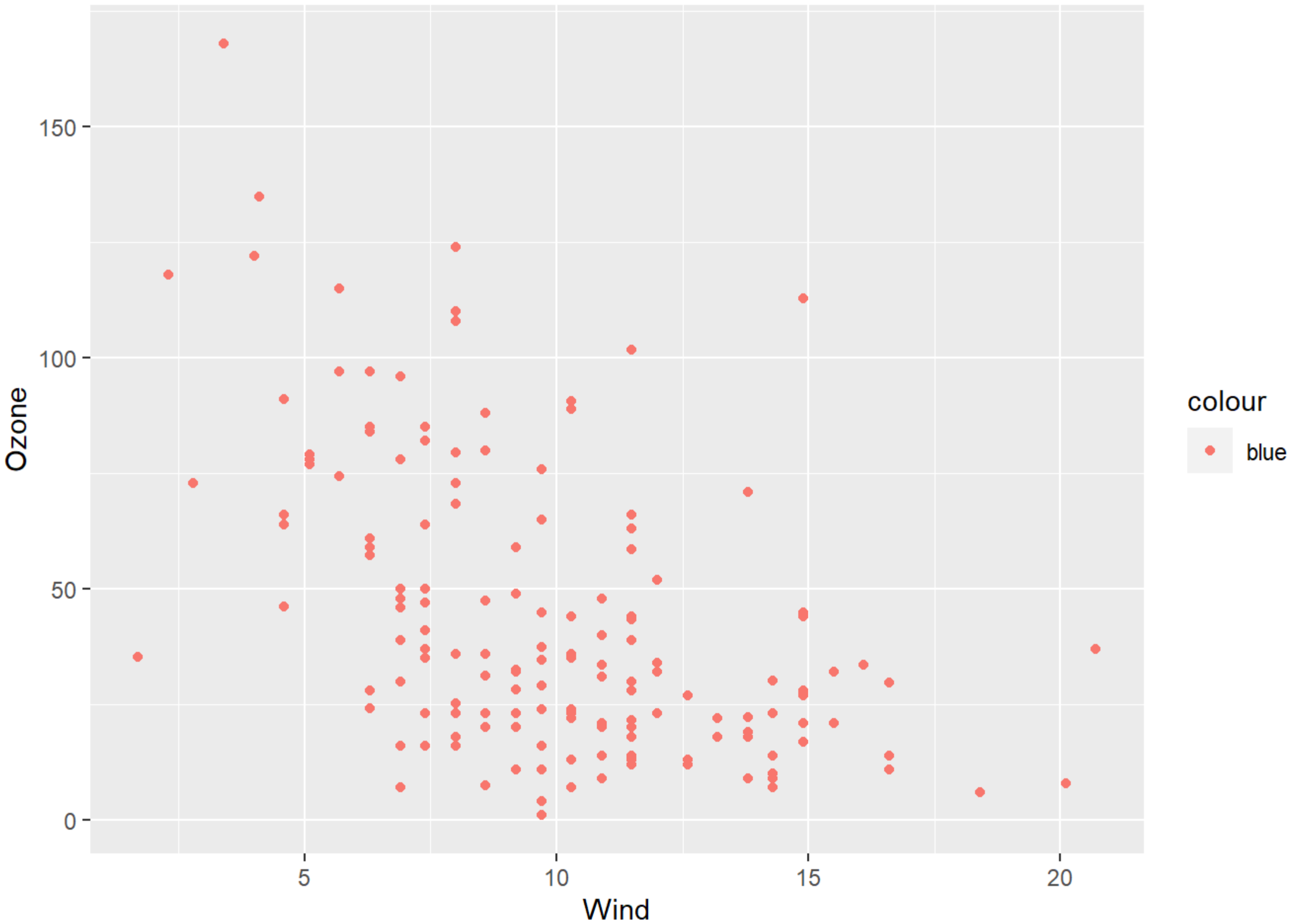
```
## [1] Other columns did not have any missing data as seen in the prior checking in "Part C".
```

E. Create **3 bivariate scatterplots (X-Y) plots** (using ggplot), for each of the predictors with the outcome. **Hint:** In each case, put **Ozone on the Y-axis**, and a **predictor on the X-axis**. Add a comment to each, describing the plot and explaining whether there appears to be a **linear relationship** between the outcome variable and the respective predictor.

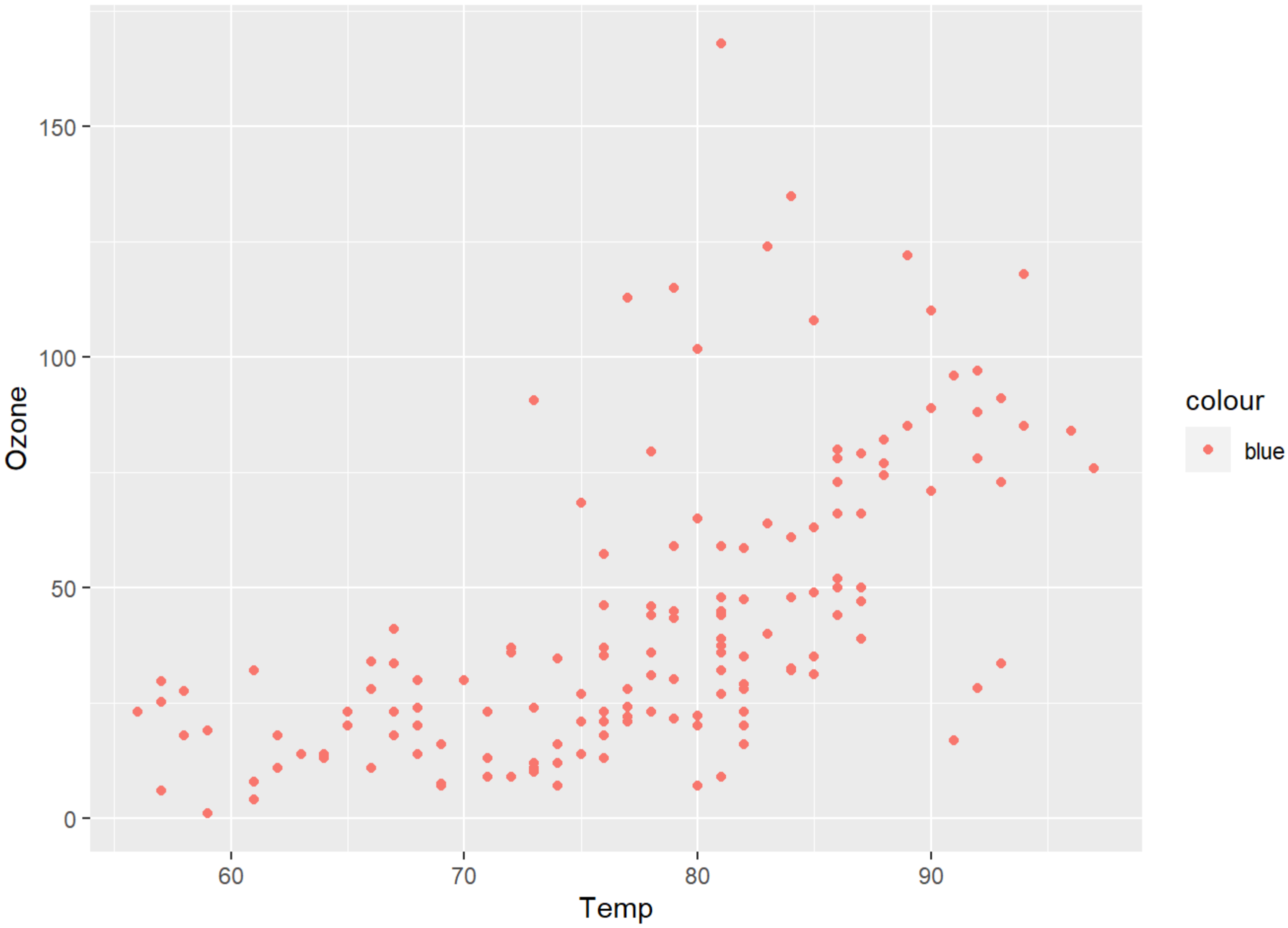
```
#   CREATE 3 BIVARIATE SCATTER PLOTS
#plot1 <- ggplot(air)
plot1 <- ggplot() + geom_point(data = air, aes(y = Ozone, x = Solar.R, color = 'blue'))
#   Does not appear to have a "linear relationship"
plot1
```




```
plot2 <- ggplot() + geom_point(data = air, aes(y = Ozone, x = Wind, color = 'blue'))
#   Does appear to have a "linear relationship" with a negative slope.
plot2
```




```
plot3 <- ggplot() + geom_point(data = air, aes(y = Ozone, x = Temp, color = 'blue'))
#   Does appear to have a "linear relationship" with a positive slope.
plot3
```




```
#ap.L <- map.L + geom_polygon(color = "black", aes(x = long, y = lat, group = group, fill = su
mPopulation))
#map.L <- map.L + coord_map()
#Solar.R + Wind + Temp
```

F. Next, create a **simple regression model** predicting **Ozone based on Wind**, using the **lm( )** command. In a comment, report the **coefficient** (aka **slope** or **beta weight**) of **Wind** in the regression output and, **if it is statistically significant**, **interpret it** with respect to **Ozone**. Report the **adjusted R-squared** of the model and try to explain what it means.

```
#   CREATE MODEL:
modelF <- lm(formula = Ozone ~ Wind, data = air)

#   DISPLAY SUMMARY:
summary(modelF)
```

```
##
## Call:
## lm(formula = Ozone ~ Wind, data = air)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.332 -18.332  -4.155  14.163  94.594
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  89.0205     6.6991  13.288  < 2e-16 ***
## Wind         -4.5925     0.6345  -7.238 2.15e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.56 on 151 degrees of freedom
## Multiple R-squared:  0.2576, Adjusted R-squared:  0.2527
## F-statistic: 52.39 on 1 and 151 DF,  p-value: 2.148e-11
```

```
#   EXPLANATION:
#-----------------------------------------------------------------------------------------------
------------
#-  Coefficient Value:         -4.5925.

#-  Coefficient significance:  The p-value of the coefficient is less than 0.05 so it is stati
stically significant.

#-  Coefficient Meaning:       Since the coefficient is significant this means as the wind inc
reases by 1 the Ozone decreases by -4.5925

#-  Adjusted R-squared:        Value 25.27%.  This says that 25.27% of the variance of the out
come variable (ozone) can be explained with the predictor variable (Wind)
```

G. Create a **multiple regression model** predicting **Ozone** based on **Solar.R**, **Wind**, and **Temp**. **Make sure to include all three predictors in one model – NOT three different models each with one predictor.**

```
#   CREATE THE MODEL:
modelG <- lm(formula = Ozone ~ Solar.R + Wind + Temp, data = air)

#   DISPLAY SUMMARY:
summary(modelG)
```

```
##
## Call:
## lm(formula = Ozone ~ Solar.R + Wind + Temp, data = air)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.651 -15.622  -4.981  12.422 101.411
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -52.16596   21.90933  -2.381   0.0185 *
## Solar.R       0.01654    0.02272   0.728   0.4678
## Wind         -2.69669    0.63085  -4.275 3.40e-05 ***
## Temp          1.53072    0.24115   6.348 2.49e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.26 on 149 degrees of freedom
```

```
## Multiple R-squared:  0.4321, Adjusted R-squared:  0.4207
## F-statistic: 37.79 on 3 and 149 DF,  p-value: < 2.2e-16
```

H. Report the **adjusted R-Squared** in a comment – how does it compare to the adjusted R-squared from Step F? Is this better or worse? Which of the predictors are **statistically significant** in the model? In a comment, report the coefficient of each predictor that is statistically significant. Do not report the coefficients for predictors that are not significant.

```
#   The adjusted R-Squared value on the multiple regression model (part g) is 42.07%  This is significantly higher than the simple regression model (part f) of only 25.76%.  This means that the multiple regression model explains more of the variance in the outcome variable than the simple regression model does.

#  The predictors that are statistically significantly are:
#   - 'Wind'
#   - 'Temp'
#* These have a p-value of less than 0.05.  The intercept is required in all cases.  In this case the intercept p-value is less than 0.05 but even if it was higher we would still have to include it in the model.
```

I. Create a one-row data frame like this:

```
#   CREATE ONE-ROW DATA FRAME:
predDF <- data.frame(Solar.R=290, Wind=13, Temp=61)
```

and use it with the **predict( )** function to predict the **expected value of Ozone**:

```
#   USE PREDICTION FUNCTION:
prediction <- predict(modelG, predDF, type = 'response')

#   PRINT RESULTS:
print(prediction)
```

```
##       1
## 10.9464
```

J. Create an additional **multiple regression model**, with **Temp** as the **outcome variable**, and the other **3 variables** as the **predictors**.

Review the quality of the model by commenting on its **adjusted R-Squared**.

```
#   CREATE THE MODEL:
modelJ <- lm(formula = Temp ~ Solar.R + Wind + Ozone, data = air)

#   DISPLAY SUMMARY:
summary(modelJ)
```

```
##
## Call:
## lm(formula = Temp ~ Solar.R + Wind + Ozone, data = air)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -18.831  -4.802   1.174   4.880  18.004 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 74.693222   2.796787  26.707  < 2e-16 ***
## Solar.R      0.015751   0.006737   2.338  0.02072 *  
## Wind        -0.580176   0.195774  -2.963  0.00354 ** 
## Ozone        0.139055   0.021907   6.348 2.49e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 7.313 on 149 degrees of freedom
## Multiple R-squared:  0.4148, Adjusted R-squared:  0.403 
## F-statistic: 35.21 on 3 and 149 DF,  p-value: < 2.2e-16
```

```
#   The adjusted R-Squared value is 40.3%.  This means that 40.3% of the variance in 'Temp' ca
n be explained using the 3 predictor variables of 'Solar.R', 'Wind', and 'Ozone'.  All of the
predictor variables are statistically significant because their respective p-values are each l
ess than 0.05.
```