Assignment 3: Predicting insurance charges by age and BMI

Felicia Liu

09/11/2020

Instructions

- Solutions will be released on Tuesday, September 15.
- This semester, homework assignments are for practice only and will not be turned in for marks.

Helpful hints:

- Every function you need to use was taught during lecture! So you may need to revisit the lecture code to help you along by opening the relevant files on Datahub. Alternatively, you may wish to view the code in the condensed PDFs posted on the course website. Good luck!
- Knit your file early and often to minimize knitting errors! If you copy and paste code for the slides, you are bound to get an error that is hard to diagnose. Typing out the code is the way to smooth knitting! We recommend knitting your file each time after you write a few sentences/add a new code chunk, so you can detect the source of knitting errors more easily. This will save you and the GSIs from frustration! **You must knit correctly before submitting.**
- It is good practice to not allow your code to run off the page. To avoid this, have a look at your knitted PDF and ensure all the code fits in the file. If it doesn't look right, go back to your .Rmd file and add spaces (new lines) using the return or enter key so that the code runs onto the next line.

```r
library(readr)
library(dplyr)
library(ggplot2)
library(broom)
library(forcats)
```

Predicting insurance charges by age and BMI

Problem: Medical insurance charges can vary according to the complexity of a procedure or condition that requires medical treatment. You are tasked with determining how these charges are associated with age, for patients who have a body mass index (bmi) in the "normal" range (bmi between 16 and 25) who are smokers.

Plan: You have chosen to use tools to examine relationships between two variables to address the problem. In particular, scatter plots and simple linear regression.

Data: You have access to the dataset `insurance.csv`, a claims dataset from an insurance provider.

Analysis and Conclusion: In this assignment you will perform the analysis and make a conclusion to help answer the problem statement.

1. [1 point] Please type one line of code below to import these data into R. Assign the data to `insure_data`. Execute the code by hitting the green arrow and ensure the data set has been saved by looking at the environment tab and viewing the data set by clicking the table icon to the right of its name.

```
insure_data <- read_csv('insurance.csv')
```

```
## Parsed with column specification:
## cols(
##   age = col_double(),
##   sex = col_character(),
##   bmi = col_double(),
##   children = col_double(),
##   smoker = col_character(),
##   region = col_character(),
##   charges = col_double()
## )
```

```
# ------- REMINDER -------
# The checks on this homework are only provided as sanity checks.
# They do not guarantee correctness.
# ------------------------

check_problem1()
```

```
## [1] "Checkpoint 1 Passed: You've loaded the dataset."
## [1] "Checkpoint 2 Passed: The data format is correct."
##
## Problem 1
## Checkpoints Passed: 2
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

Use the space below to use the functions from lecture to get to know your dataset. Execute these functions line by line so you can look at their output, and examine these data.

```
dim(insure_data)
```

```
## [1] 1338    7
```

```
names(insure_data)
```

```
## [1] "age"      "sex"      "bmi"      "children" "smoker"   "region"   "charges"
```

```
str(insure_data)
```

```
## tibble [1,338 x 7] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ age     : num [1:1338] 19 18 28 33 32 31 46 37 37 60 ...
##  $ sex     : chr [1:1338] "female" "male" "male" "male" ...
##  $ bmi     : num [1:1338] 27.9 33.8 33 22.7 28.9 ...
##  $ children: num [1:1338] 0 1 3 0 0 0 1 3 2 0 ...
##  $ smoker  : chr [1:1338] "yes" "no" "no" "no" ...
##  $ region  : chr [1:1338] "southwest" "southeast" "southeast" "northwest" ...
##  $ charges : num [1:1338] 16885 1726 4449 21984 3867 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   age = col_double(),
##   ..   sex = col_character(),
##   ..   bmi = col_double(),
##   ..   children = col_double(),
##   ..   smoker = col_character(),
##   ..   region = col_character(),
##   ..   charges = col_double()
##   .. )
```

```
head(insure_data)
```

```
## # A tibble: 6 x 7
##     age sex      bmi children smoker region    charges
##   <dbl> <chr>  <dbl>    <dbl> <chr>  <chr>       <dbl>
## 1    19 female  27.9        0 yes    southwest  16885.
## 2    18 male    33.8        1 no     southeast   1726.
## 3    28 male    33          3 no     southeast   4449.
## 4    33 male    22.7        0 no     northwest  21984.
## 5    32 male    28.9        0 no     northwest   3867.
## 6    31 female  25.7        0 no     southeast   3757.
```

2. [1 point] How many individuals are in the dataset? Assign this number to p2.

```
p2 <- 1338


check_problem2()
```

```
## [1] "Checkpoint 1 Passed: p2 is correctly assigned as a numeric."
## [1] "Checkpoint 2 Passed: You've got the correct number of individuals."
##
## Problem 2
## Checkpoints Passed: 2
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

3. [1 point] What are the nominal variables in the dataset? Assign the names of these variables to a vector of strings, p3.

```
p3 <- c("sex", "smoker", "region")


check_problem3()
```

```
## [1] "Checkpoint 1 Passed: p3 is correctly assigned as a vector."
## [1] "Checkpoint 2 Passed: Variables are correctly specified as strings."
## [1] "Checkpoint 3 Passed: The required variables are included."
## [1] "Checkpoint 4 Passed: The length of p3 is correct."
##
## Problem 3
## Checkpoints Passed: 4
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

4. [1 point] How many ordinal variables are in the dataset? Assign the number of ordinal variables to p4.

```
p4 <- 0


check_problem4()
```

```
## [1] "Checkpoint 1 Passed: p4 is correctly assigned as numeric."
## [1] "Checkpoint 2 Passed: The number of selected variables are correct."
##
## Problem 4
## Checkpoints Passed: 2
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

5. [1 point] Are there continuous variables in the dataset? Assign the names of these variables to a vector of strings, p5.

```r
p5 <- c("age", "bmi", "charges")


check_problem5()
```

```
## [1] "Checkpoint 1 Passed: p5 is assigned as a vector."
## [1] "Checkpoint 2 Passed: Variable names are entered as strings."
##
## Problem 5
## Checkpoints Passed: 2
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

6. [1 point] What are the discrete variables in the dataset? Assign the names of these variables to a vector of strings, p6.

```r
p6 <- c("children")


check_problem6()
```

```
## [1] "Checkpoint 1 Passed: p6 is assigned to a vector of strings."
## [1] "Checkpoint 2 Passed: You've choosen the correct variables."
##
## Problem 6
## Checkpoints Passed: 2
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

Run the following code by removing the "#" symbol in front of each of the six lines and executing the code chunk. Remind yourself what the `mutate()` function does in general, and notice that a new function `case_when()` is also being used.

##STOP: Remove eval = F from this chunk header before proceeding

```
insure_data <- insure_data %>%
  mutate(bmi_cat = case_when(bmi < 16 ~ "Underweight",
                             bmi >= 16 & bmi < 25 ~ "Normal",
                             bmi >= 25 & bmi < 30 ~ "Overweight",
                             bmi >= 30 ~ "Obese")
                  )
```

7. [1 point] What between 16did the above code achieve?:

The code adds a new column to the data that categorizes whether the individual is "Underweight," "Normal," "Overweight," and "Obese" based on various BMI ranges. The new column would show "Underweight" if the BMI is under 16, "Normal" if the BMI was equal to 16 and under 25, "Overweight" if the BMI was greater than or equal to 25 and less than 30, and "Obese" if the BMI was over 30.

8. [1 point] What type of variable is `bmi_cat`? Uncomment one of the choices below.

```
p8 <- "ordinal"
# p8 <- "nominal"
# p8 <- "continuous"
# p8 <- "discrete"

# This only checks that you've selected an answer, not its correctness.
check_problem8()
```

```
## [1] "Checkpoint 1 Passed: Correct choice."
##
## Problem 8
## Checkpoints Passed: 1
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

9. [1 point] Read the problem statement proposed at the beginning of this exercise. Who belongs to the population of interest? Uncomment one of the choices below.

```
p9 <- "Smokers of normal BMI"
# p9 <- "Smokers of overweight BMI"
# p9 <- "Smokers who have abnormal BMI"
# p9 <- "All people at risk of high medical charges"

# This only checks that you've selected an answer, not its correctness.
check_problem9()
```

```
## [1] "Checkpoint 1 Passed: p9 is set to one of the provided choices."
##
## Problem 9
## Checkpoints Passed: 1
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

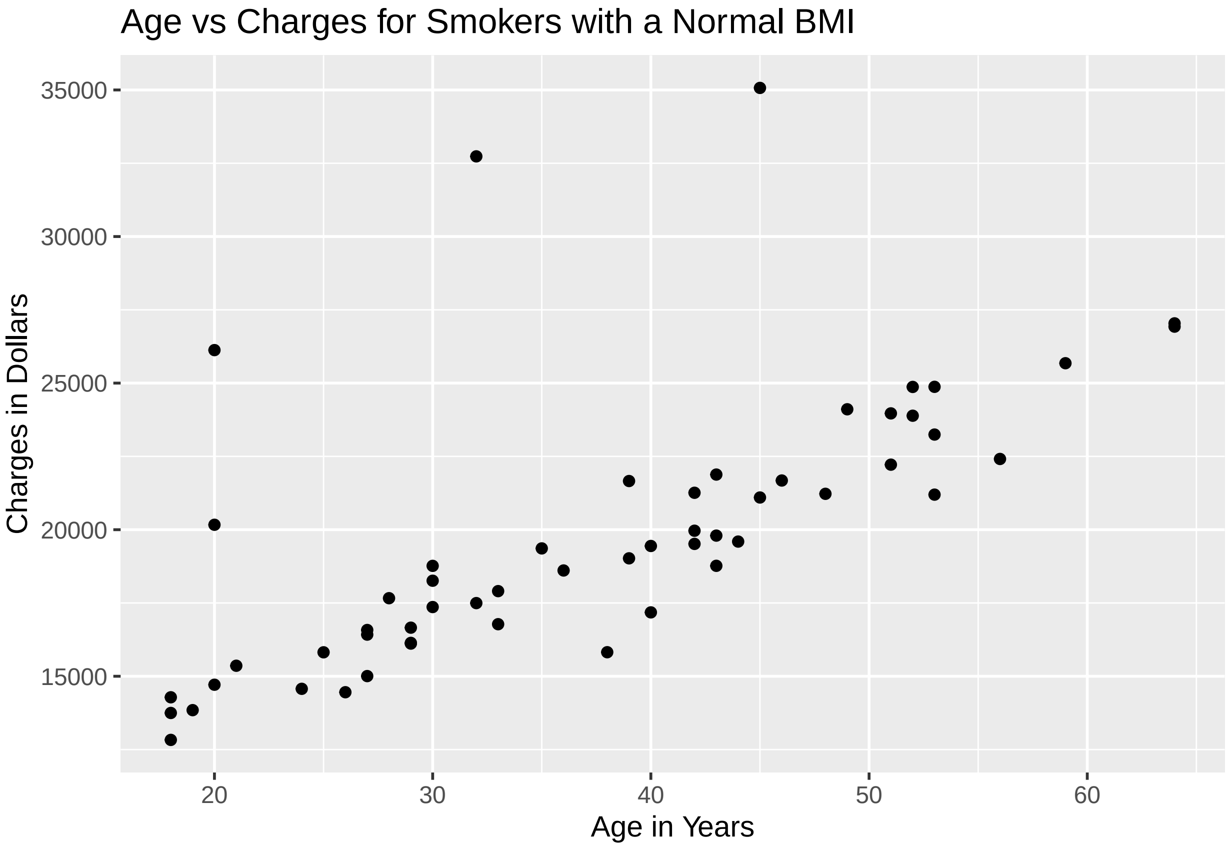
10. [1 point] Using a dplyr function, make a new dataset called `insure_subset` containing the population of interest.

```
insure_subset <- insure_data %>% filter(smoker == "yes" & bmi >= 16 & bmi < 25)


check_problem10()
```

```
## [1] "Checkpoint 1 Passed: Insure_subset is built as a data frame."
## [1] "Checkpoint 1 Passed: The number of observation lies in the current range."
##
## Problem 10
## Checkpoints Passed: 2
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

11. [3 points] Make a scatter plot of the relationship between age and insurance charges for the population of interest. Give your plot an informative title.

```
p11 <- ggplot(insure_subset, aes(x = age, y = charges)) + geom_point() +
  labs(title = "Age vs Charges for Smokers with a Normal BMI", x = "Age in Years",
  y = "Charges in Dollars")
p11
```



```
check_problem11()
```

```
## [1] "Checkpoint 1 Passed: You’ve defined a ggplot."
## [1] "Checkpoint 2 Passed: You’ve use the right dataset."
## [1] "Checkpoint 3 Passed: You’ve choosen the correct variable for x axis."
## [1] "Checkpoint 4 Passed: You’ve choosen the correct variable for y axis."
## [1] "Checkpoint 5 Passed: You’ve used the scatter plot."
## [1] "Checkpoint 6 Passed: You’ve added a title."
##
## Problem 11
## Checkpoints Passed: 6
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

12. [2 points] Run a linear regression model on the relationship between `age` and `charges`. Think about which variable is explanatory (X) and which is response (Y). Assign the regression model to the name `insure_mod`. Then type `tidy(insure_mod)` below the model's code and execute both lines.

```
insure_model <- lm(charges ~ age, insure_subset)
tidy(insure_model)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic       p.value
##   <chr>          <dbl>     <dbl>     <dbl>         <dbl>
## 1 (Intercept)   10656.    1471.       7.24 0.00000000184
## 2 age             246.      37.4      6.58 0.0000000217
```

```
check_problem12()
```

```
## [1] "Checkpoint 1 Passed: You've fit a linear regression model."
## [1] "Checkpoint 2 Passed: A variable is included correctly."
## [1] "Checkpoint 3 Passed: A variable is included correctly."
##
## Problem 12
## Checkpoints Passed: 3
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

13a. [1 point] Interpret the slope parameter:

The slope parameter is 246, which means that for every 1 year increase in age, the medical insurance charges of smokers with a normal BMI increases by $246.

13b. [1 point] Interpret the intercept parameter:

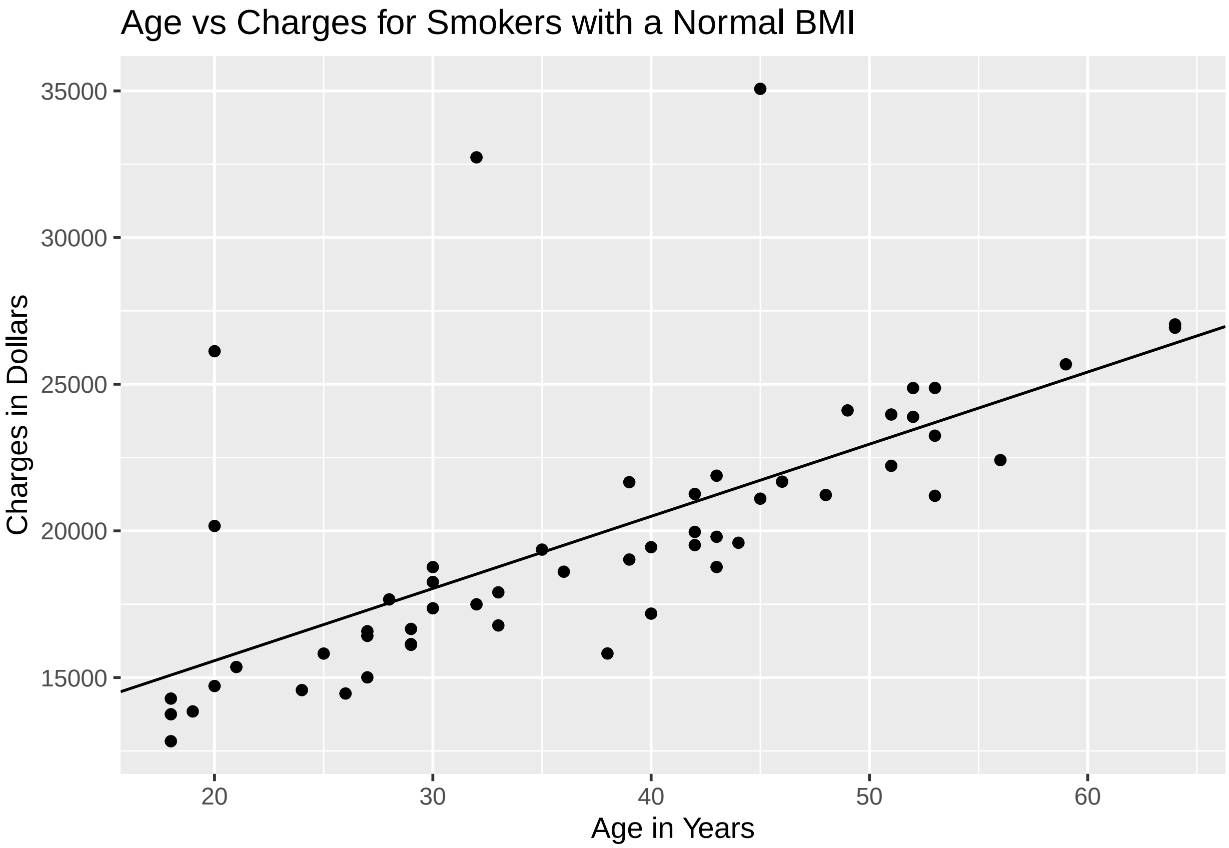
The intercept parameter is 10656, which means that for a smoker with a normal BMI, at age 0 the charge for medical insurance would be $10656.

13c. [1 point] Does the intercept make sense in this context?:

The intercept does not make sense in this context because at age 0, it is highly likely that an individual would be a smoker, have a normal BMI, and rack up $10656 in medical insurance expenses altogether. In this situation, a linear regression model would not be accurate for the extreme endpoints of the scatterplot.

14. [1 point] Add the line of best fit to your scatterplot by copying and pasting the plot's code from Problem 11 into the chunk below and adding a `geom` that can be used to add a regression line:

```
p14 <- ggplot(insure_subset, aes(x = age, y = charges)) + geom_point() +
  labs(title = "Age vs Charges for Smokers with a Normal BMI", x = "Age in Years",
  y = "Charges in Dollars") + geom_abline(intercept = 10656, slope = 246)
p14
```



```
check_problem14()
```

```
## [1] "Checkpoint 1 Passed: You've defined a ggplot."
## [1] "Checkpoint 2 Passed: You used the right dataset."
## [1] "Checkpoint 3 Passed: You've selected the correct variable for x axis."
## [1] "Checkpoint 4 Passed: You've selected the correct variable for y axis."
## [1] "Checkpoint 5 Passed: You've used the scatter plot."
## [1] "Checkpoint 6 Passed: You've added a title."
## [1] "Checkpoint 7 Passed: You've plotted the regression line."
##
## Problem 14
## Checkpoints Passed: 7
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

15. [2 points] What do you notice about the fit of the line in terms of the proportion of points above vs. below the line. Why do you think that is?:

There seems to be more points under the line compared to above the line. I think this is the case because there are high outliers in the data set. There are individuals with extremely high medical insurance charges, and basically the line of best fit gets pulled towards those points.

Run the following `filter()` function by removing the "#" symbol in front of the two lines of code and executing the code chunk.

STOP: Remove eval = F from this chunk header before proceeding

```
insure_smaller_subset <- insure_subset %>%
  filter(charges < 30000 & ! (charges > 25000 & age == 20))
```

16. [2 points] How many individuals were removed? Who were they?:

3 individuals were removed. They were the two points with charges over $30000 and the individual aged 20 with a charge of over $25000.

17. [2 points] Run a regression model on `insure_smaller_subset` between `charges` and `age`. Assign it to insure_better_model and look at the output using the `tidy()` function, as was done with the previous linear model.

```
insure_better_model <- lm(charges ~ age, insure_smaller_subset)
tidy(insure_better_model)
```

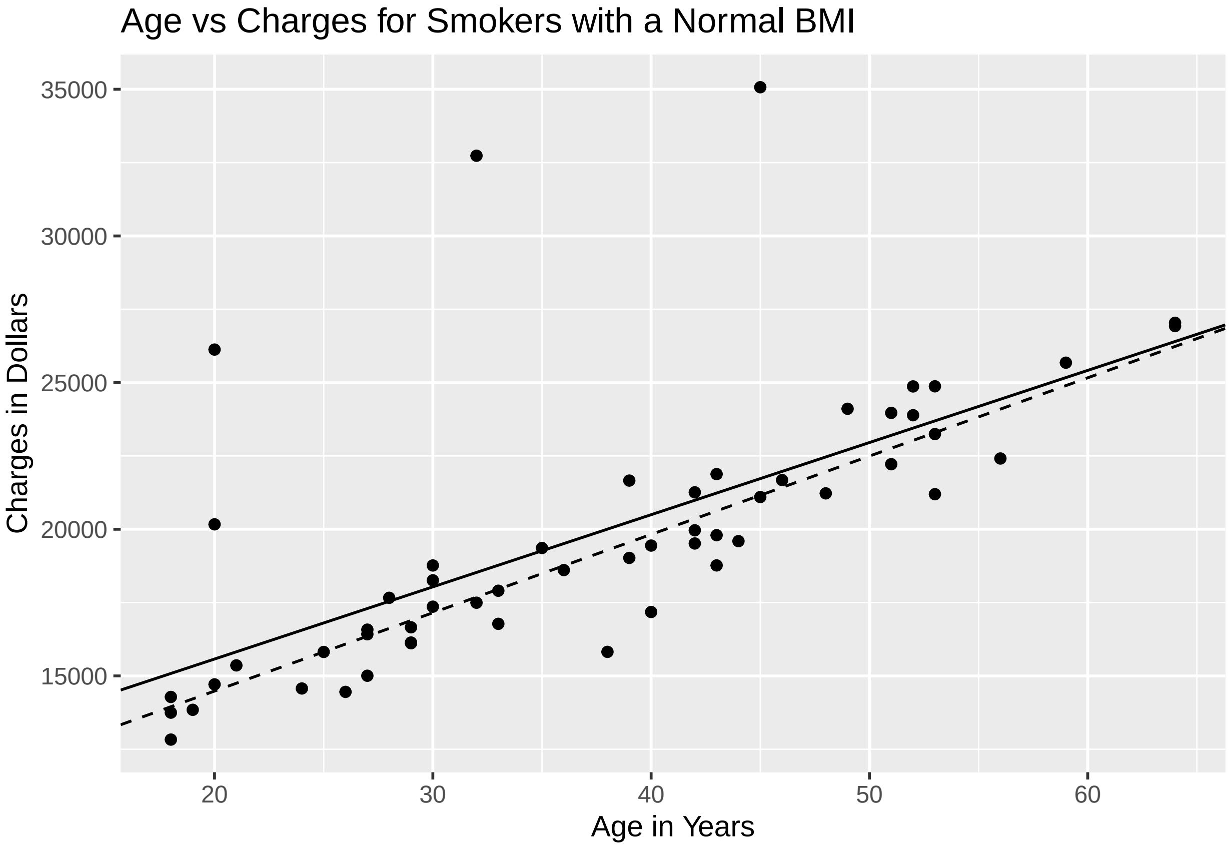
```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    9144.     633.       14.4 1.81e-19
## 2 age             267.      16.0      16.7 4.44e-22
```

```
check_problem17()
```

```
## [1] "Checkpoint 1 Passed: You've used the linear regression model."
## [1] "Checkpoint 2 Passed: You've included the required variables."
## [1] "Checkpoint 3 Passed: You've included the required variables."
## [1] "Checkpoint 4 Passed: The number of observations is correct"
##
## Problem 17
## Checkpoints Passed: 4
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

18. [2 points] Add the new regression line to your ggplot from Problem 14. Keep the older regression line on the plot for comparison. To distinguish them, change the color, line type, or line width of the newly-added line.

```r
p18 <- ggplot(insure_subset, aes(x = age, y = charges)) + geom_point() +
  labs(title = "Age vs Charges for Smokers with a Normal BMI", x = "Age in Years",
  y = "Charges in Dollars") + geom_abline(intercept = 10656, slope = 246) +
  geom_abline(intercept = 9144, slope = 267, lty = 2)
p18
```



```r
check_problem18()
```

```
## [1] "Checkpoint 1 Passed: You've defined a ggplot."
## [1] "Checkpoint 2 Passed: You've used the right dataset."
## [1] "Checkpoint 3 Passed: You've used the correct variable for x axis."
## [1] "Checkpoint 4 Passed: You've used the correct variable for y axis."
## [1] "Checkpoint 5 Passed: You've used a scatter plot."
## [1] "Checkpoint 6 Passed: You've added a title."
## [1] "Checkpoint 7 Passed: You've added the original regression line."
## [1] "Checkpoint 8 Passed: You've added the new regression line."
##
## Problem 18
## Checkpoints Passed: 8
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

19. [1 point] Calculate the r-squared value for `insure_model` using a function learned in class. Assign this value to insure_model_r2.

```
insure_model_r2 <- glance(insure_model) %>% pull(r.squared)


check_problem19()
```

```
## [1] "Checkpoint 1 Passed: You've assigned a numeric to insure_model_r2."
## [1] "Checkpoint 2 Passed: This lies in the correct range for a r-squared value."
## [1] "Checkpoint 3 Passed: Correct value."
##
## Problem 19
## Checkpoints Passed: 3
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

20. [1 point] Calculate the r-squared value for `insure_better_model` using a function learned in class. Assign this value to insure_better_model_r2.

```
insure_better_model_r2 <- glance(insure_better_model) %>% pull(r.squared)


check_problem20()
```

```
## [1] "Checkpoint 1 Passed: You've assigned a numeric to insure_better_model_r2."
## [1] "Checkpoint 2 Passed: This is a valid r-square value."
## [1] "Checkpoint 3 Passed: Correct value."
##
## Problem 20
## Checkpoints Passed: 3
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

21. [2 points] Calculate the correlation between `age` and `charges` using the subset `insure_subset`. Also calculate correlation squared. You should use `summarize()` and name the two new columns `corr` and `corr_sq`. What do you notice about the relationship between the correlation and r-squared values that you calculated earlier?

```r
p21 <- insure_subset %>% summarize(corr = cor(age, charges), corr_sq = corr^2)


check_problem21()
```

```
## [1] "Checkpoint 1 Passed: You've assigned a numeric to insure_better_model_r2."
## [1] "Checkpoint 2 Passed: Your input is a double variable."
## [1] "Checkpoint 3 Passed: Your input is a double variable."
##
## Problem 21
## Checkpoints Passed: 3
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

22. [2 points] Calculate the correlation between `age` and `charges` using the smaller dataset `insure_smaller_subset`. Also calculate correlation squared. You should use `summarize()` and name the two new columns `corr` and `corr_sq`. What do you notice about the relationship between the correlation and r-squared values that you calculated earlier?

```r
p22 <- insure_smaller_subset %>% summarize(corr = cor(age, charges), corr_sq = corr^2)


check_problem22()
```

```
## [1] "Checkpoint 1 Passed: You've assigned a numeric to insure_better_model_r2."
## [1] "Checkpoint 2 Passed: Your input is a double variable."
## [1] "Checkpoint 3 Passed: Your input is a double variable."
##
## Problem 22
## Checkpoints Passed: 3
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

PART B

Your supervisor asks you to extend your analysis to consider other smokers with BMIs classified as overweight or obese. In particular, she wanted to know if the relationship between age and medical charges is different for different BMI groups. You can use data visualization coupled with your skills in linear regression to help answer this question.

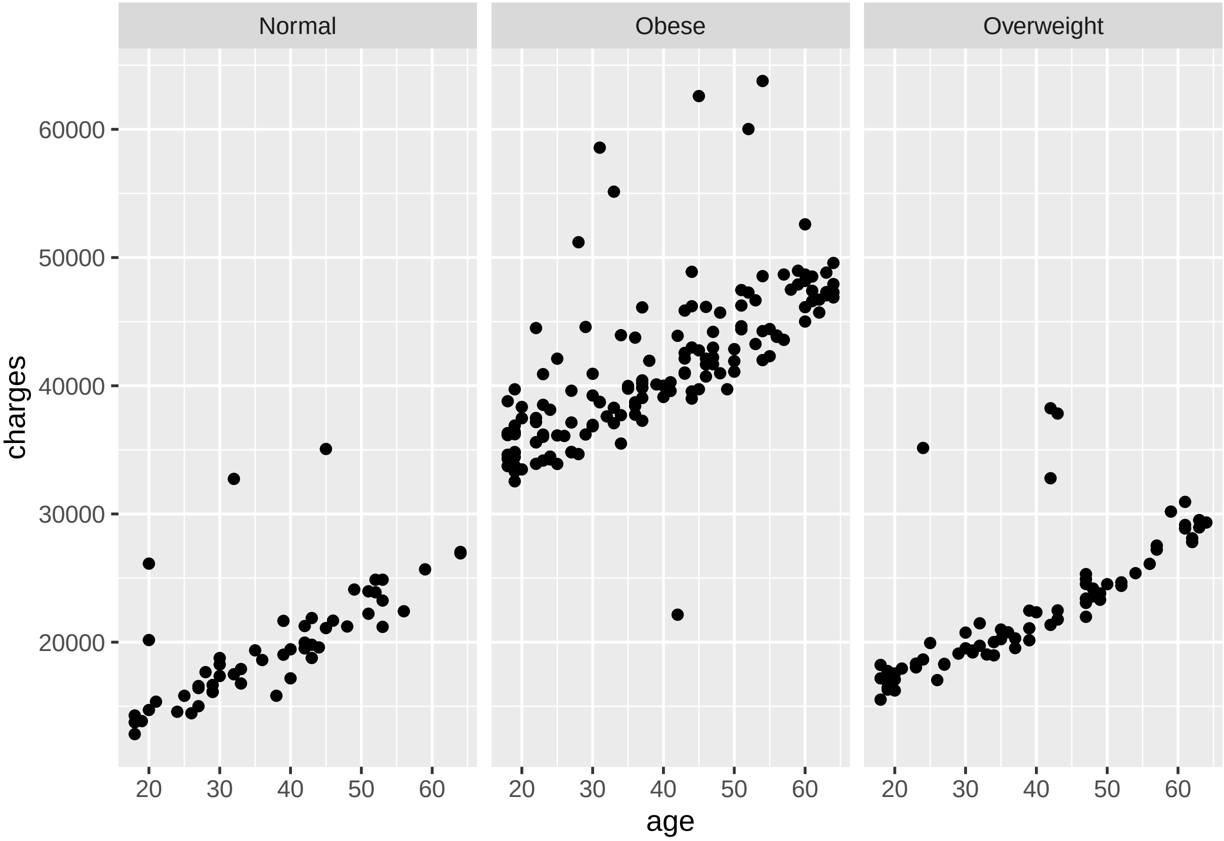
23. [1 point] **Make a new dataset called `insure_smokers` that includes smokers of any BMI.**

```
insure_smokers <- insure_data %>% filter(smoker == "yes")



check_problem23()
```

```
## [1] "Checkpoint 1 Passed: The result is a data frame."
## [1] "Checkpoint 1 Passed: The number of observations falls in the correct range."
##
## Problem 23
## Checkpoints Passed: 2
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

24. [1 point] Make a scatter plot that examines the relationship between `age` and `charges` separately for normal, overweight, and obese individuals. A `facet_` command may help you.

```
p24 <- ggplot(insure_smokers, aes(x = age, y = charges)) + geom_point() + facet_wrap(~ bmi_cat)
p24
```



```
check_problem24()
```

```
## [1] "Checkpoint 1 Passed: You've defined a ggplot."
## [1] "Checkpoint 2 Passed: You've used the right dataset."
## [1] "Checkpoint 3 Passed: Correct x variable."
## [1] "Checkpoint 4 Passed: Correct y variable."
## [1] "Checkpoint 5 Passed: You've used the scatter plot."
## [1] "Checkpoint 6 Passed: You've wrapped the variables correctly."
##
## Problem 24
## Checkpoints Passed: 6
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

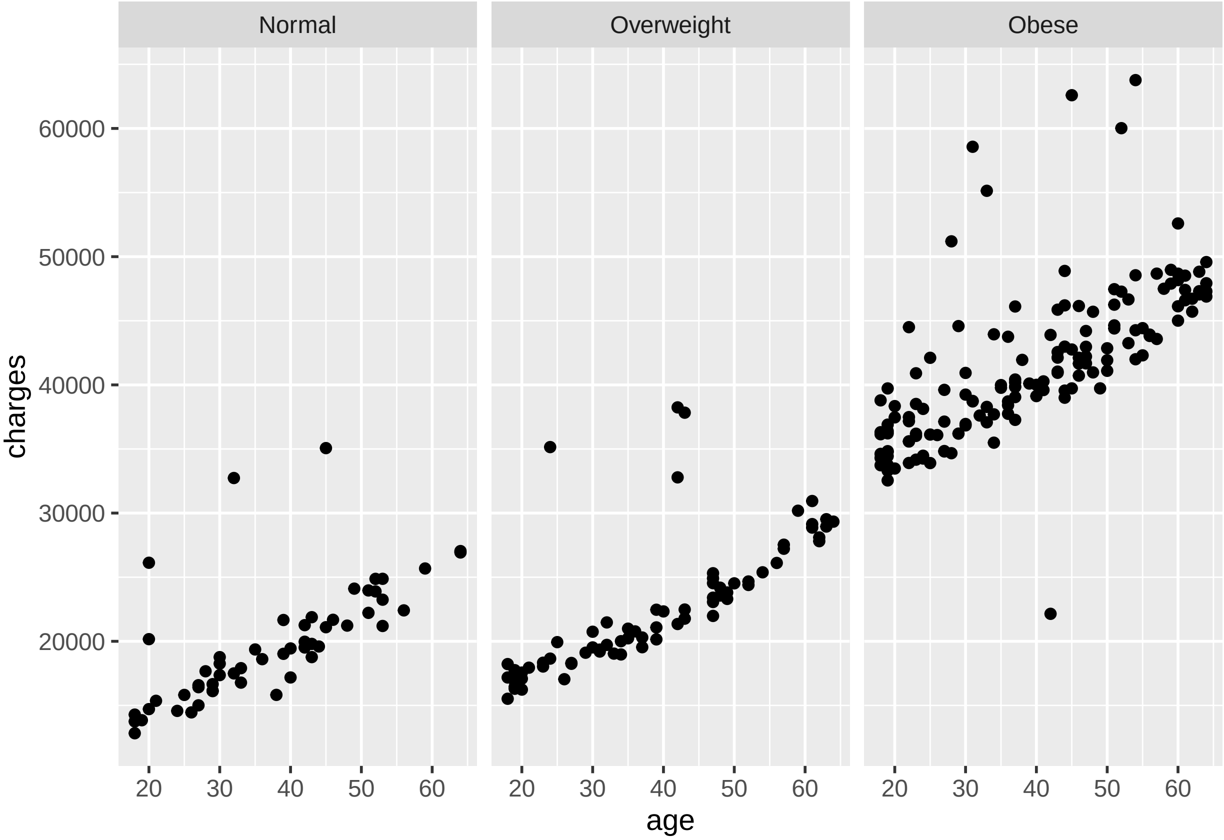
Is there something out of order with your plot you just made? The issue is that the plot is automatically displayed by listing the BMI categories alphabetically. Uncomment and run the following code chunk to assign an ordering to the values of `bmi_cat`.:

STOP: Remove eval = F from this chunk header before proceeding

```
insure_smokers <- insure_smokers %>%
  mutate(bmi_cat_ordered = forcats::fct_relevel(bmi_cat, "Normal", "Overweight", "Obese"))
```

25. [1 point] Re-run your plot code, but this time, facet using `bmi_cat_ordered`.

```
p25 <- ggplot(insure_smokers, aes(x = age, y = charges)) + geom_point() + facet_wrap(~ bmi_cat_ordered)
p25
```



```
check_problem25()
```

```
## [1] "Checkpoint 1 Passed: You’ve defined a ggplot."
## [1] "Checkpoint 2 Passed: You’ve used the right dataset."
## [1] "Checkpoint 3 Passed: Correct x variable."
## [1] "Checkpoint 4 Passed: Correct y variable."
## [1] "Checkpoint 5 Passed: You’ve used scatter plot."
## [1] "Checkpoint 6 Passed: You’ve wrapped the data points."
##
## Problem 25
## Checkpoints Passed: 6
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

26. [3 points] Run a separate linear model for each BMI group. To do this, you will need to subset your data into the three groups of interest first. Call your models `normal_mod`, `overweight_mod`, `obese_mod`. Use the `tidy()` function to display the output from each model.

```
normal_bmi = insure_smokers %>% filter(bmi_cat_ordered == "Normal")
overweight_bmi = insure_smokers %>% filter(bmi_cat_ordered == "Overweight")
obese_bmi = insure_smokers %>% filter(bmi_cat_ordered == "Obese")

normal_mod <- lm(charges ~ age, normal_bmi)
overweight_mod <- lm(charges ~ age, overweight_bmi)
obese_mod <- lm(charges ~ age, obese_bmi)

tidy(normal_mod)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic       p.value
##   <chr>          <dbl>     <dbl>     <dbl>         <dbl>
## 1 (Intercept)   10656.     1471.      7.24 0.00000000184
## 2 age             246.      37.4      6.58 0.0000000217
```

```
tidy(overweight_mod)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   12400.     1176.     10.5  3.01e-16
## 2 age             264.      28.9      9.16 1.07e-13
```

```
tidy(obese_mod)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   30558.     1093.      28.0 7.96e-60
## 2 age             281.      26.2      10.7 5.05e-20
```

```
check_problem26()
```

```
## [1] "Checkpoint 1 Passed: A linear regression model is used."
## [1] "Checkpoint 2 Passed: A required variable has been selected."
## [1] "Checkpoint 3 Passed: A required variable has been selected."
## [1] "Checkpoint 4 Passed: A linear model is used."
## [1] "Checkpoint 5 Passed: A required variable has been selected."
## [1] "Checkpoint 6 Passed: A required variable has been selected."
## [1] "Checkpoint 7 Passed: A linear model is used."
## [1] "Checkpoint 8 Passed: A required variable has been selected."
## [1] "Checkpoint 9 Passed: A required variable has been selected."
##
## Problem 26
## Checkpoints Passed: 9
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

For the next three problems, use the models to predict medical charges for a 20-year old by weight category. You don't need an R function to make these predictions, just the output from the model. Show your work for each calculation by writing the mathematical expression in and round to the nearest dollar.

27. [1 point] **...among normal BMI group:**

```
p27 <- 246*20 + 10656


check_problem27()
```

```
## [1] "Checkpoint 1 Passed: The result is numeric."
## [1] "Checkpoint 1 Passed: The result falls in the correct range."
##
## Problem 27
## Checkpoints Passed: 2
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

28. [1 point] **...among overweight BMI group:**

```
p28 <- 264*20 + 12400


check_problem28()
```

```
## [1] "Checkpoint 1 Passed: The result is numeric."
## [1] "Checkpoint 1 Passed: The result falls in the correct range."
##
## Problem 28
## Checkpoints Passed: 2
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

29. [1 point] **...among obese BMI group:**

```
p29 <- 281*20 + 30558


check_problem29()
```

```
## [1] "Checkpoint 1 Passed: The result is numeric."
## [1] "Checkpoint 1 Passed: The result falls in the correct range."
##
## Problem 29
## Checkpoints Passed: 2
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

30. [3 points] In three sentences maximum, (1) comment on the direction of the association, (2) comment on how much the slopes vary across the BMI groups, and (3) how much the prediction for a 20-year old varies.

There is a strong, linear, positive association between the age of the patient and the medical insurance charge. The slopes of the BMI group vary quite a bit - from 246 for normal, to 264 for overweight, and 281 for obese. The charge prediction for a 20-year old also varies drastically, from $15576 for normal BMI, $17680 for overweight BMI, and $36178 for obese BMI. The slopes and charge predictions for normal and overweight groups are closer together while those of the obese group is significantly higher.

Check your score

Click on the middle icon on the top right of this code chunk (with the downwards gray arrow and green bar) to run all your code in order. Then, run this chunk to check your score.

```
# Just run this chunk.
total_score()
```

```
##                      Test Points_Possible          Type
## Problem 1          PASSED               1    autograded
## Problem 2          PASSED               1    autograded
## Problem 3          PASSED               1    autograded
## Problem 4          PASSED               1    autograded
## Problem 5          PASSED               1    autograded
## Problem 6          PASSED               1    autograded
## Problem 7  NOT YET GRADED               1 free-response
## Problem 8          PASSED               1    autograded
## Problem 9          PASSED               1    autograded
## Problem 10         PASSED               1    autograded
## Problem 11         PASSED               3    autograded
## Problem 12         PASSED               2    autograded
## Problem 13 NOT YET GRADED               3 free-response
## Problem 14         PASSED               1    autograded
## Problem 15 NOT YET GRADED               2 free-response
## Problem 16 NOT YET GRADED               2 free-response
## Problem 17         PASSED               2    autograded
## Problem 18         PASSED               2    autograded
## Problem 19         PASSED               1    autograded
## Problem 20         PASSED               1    autograded
## Problem 21         PASSED               2    autograded
## Problem 22         PASSED               2    autograded
## Problem 23         PASSED               1    autograded
## Problem 24         PASSED               1    autograded
## Problem 25         PASSED               1    autograded
## Problem 26         PASSED               3    autograded
## Problem 27         PASSED               1    autograded
## Problem 28         PASSED               1    autograded
## Problem 29         PASSED               1    autograded
## Problem 30 NOT YET GRADED               3 free-response
```