# DATA 101 Exam 2

Kamila Palys

Due: Sunday, 11/29 at 11:59pm

## Academic Honesty Statement (fill in your name)

I, Kamila Palys, hereby affirm that I have not communicated with or gained information in any way from my classmates or anyone other than the Professor during this exam, that I have not assisted anyone else with this exam, and that all work is my own.

## Load packages and data

```
# load required packages here
library(tidyverse)
library(tidymodels)
library(janitor)
library(lubridate)
library(NHANES)
```

```
# read in the data here
nhanes <- NHANES %>%
  janitor::clean_names()
```

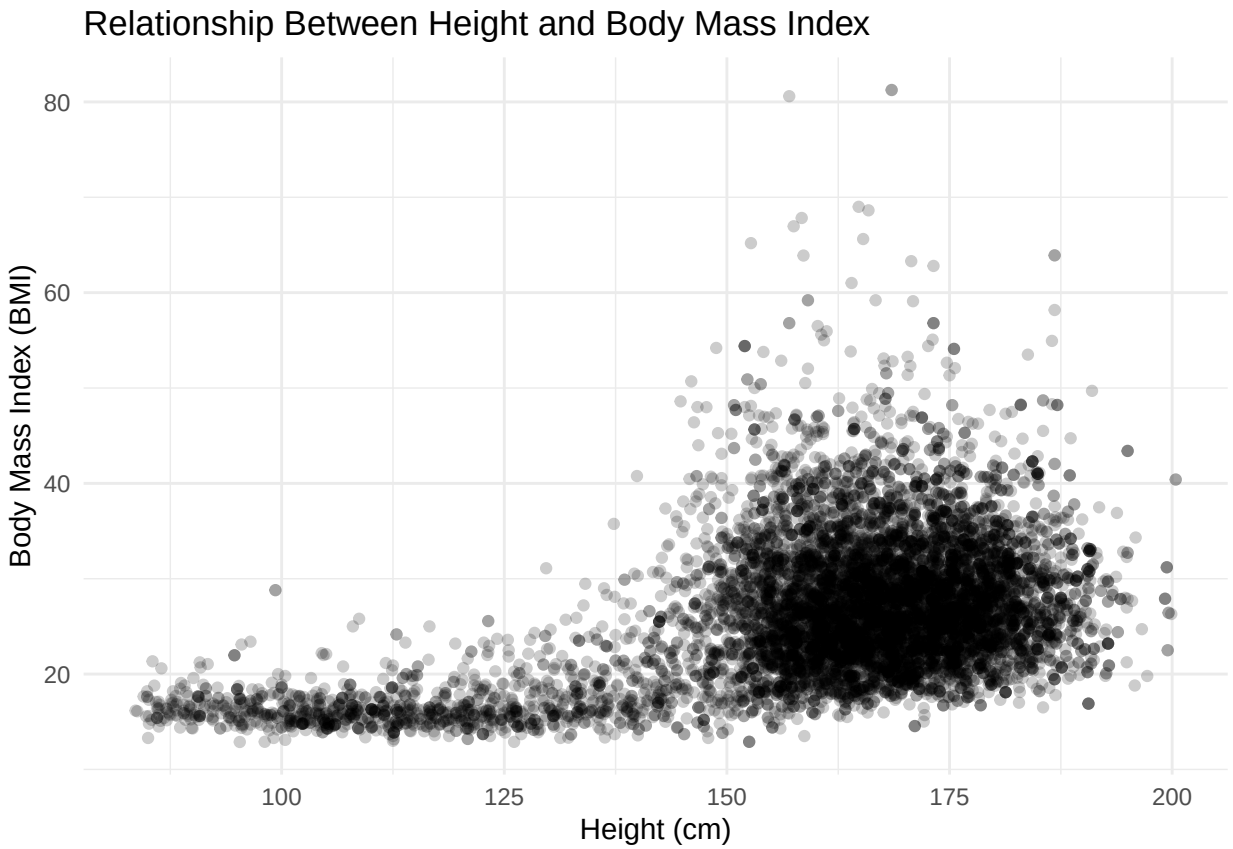
## Questions

### Question 1

Here we will show the relationship between the height and body mass index of a person using a scatterplot.

```
ggplot(data = nhanes, mapping = aes(x = height, y = bmi)) +
  geom_point(alpha = 0.2) +
  labs(x = "Height (cm)",
       y = "Body Mass Index (BMI)",
       title = "Relationship Between Height and Body Mass Index") +
  theme_minimal()
```

```
## Warning: Removed 366 rows containing missing values (geom_point).
```

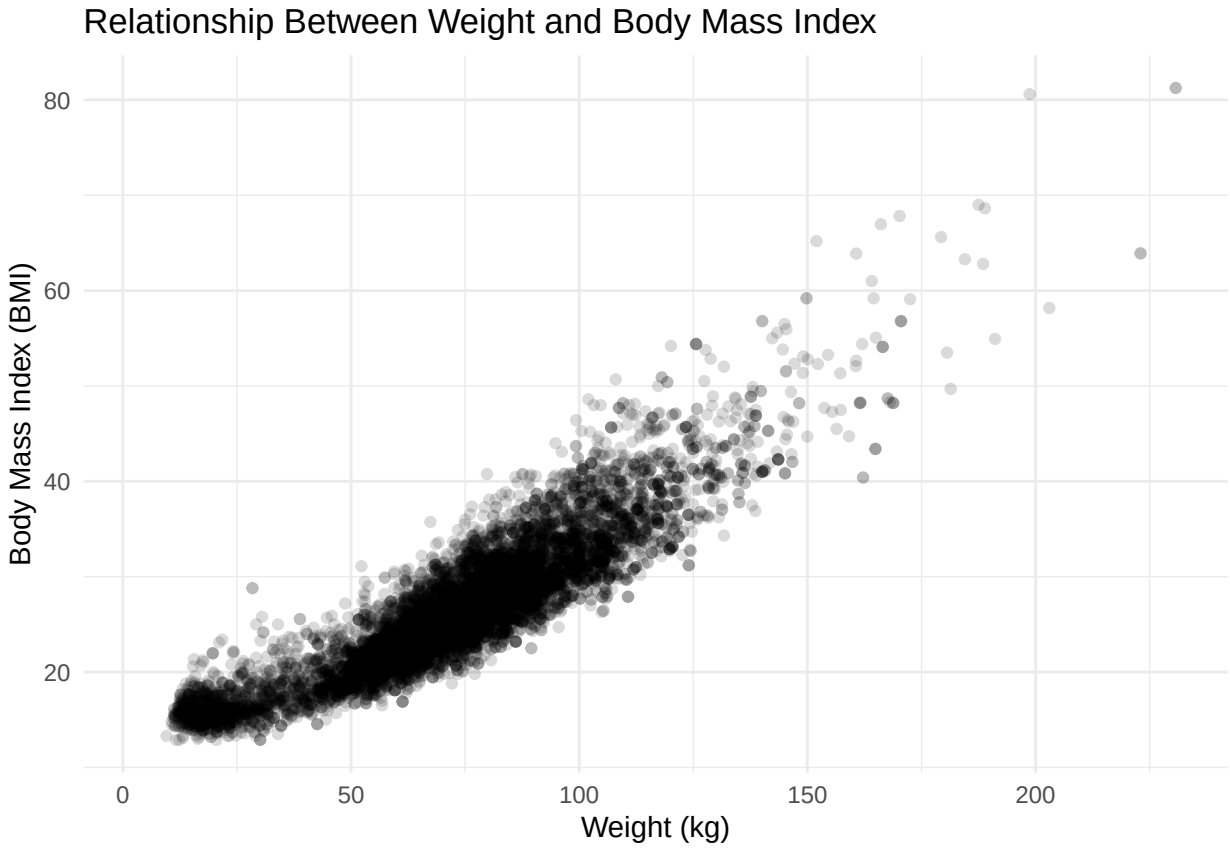



From this display, a slight upwards trend is seen. For heights of less than 150cm, not much of a slope is visible and body mass index stays relatively the same for various heights under 150cm. For heights above that amount, there are many more people whose body mass indexes are higher and there is a far greater range of them. In fact, most of the body mass indexes here are higher than for people shorter than 150cm. One reason as to why this may be the case is because those shorter than 150cm are usually young children, whose parents have the main say in their diets. When the children grow up and therefore also get taller, they make more of their own decisions when it comes to their diet, which may not be good for their health and BMI.

Now we will show the relationship between the weight and body mass index of a person, also using a scatterplot.

```
ggplot(data = nhanes, mapping = aes(x = weight, y = bmi)) +
  geom_point(alpha = 0.15) +
  labs(x = "Weight (kg)",
       y = "Body Mass Index (BMI)",
       title = "Relationship Between Weight and Body Mass Index") +
  theme_minimal()
```

```
## Warning: Removed 366 rows containing missing values (geom_point).
```




Here, there is almost a perfectly linear positive slope between a person's weight and their body mass index. The positive relationship makes sense because as the weight of a person goes up, given a height, their body mass index will go up as well. Even though height is not a factor shown on this graph, it often increases along with weight, which may lead to little to no change in body mass index. However, weight can fluctuate a lot more and throughout a person's entire life, which is why this upward trend is seen, suggesting that a higher weight still generally leads to a higher body mass index.

### Question 2

Now, a linear model will be created to predict BMI as a function of weight.

```
bmi_fit <- lm(bmi ~ weight, data = nhanes)
tidy(bmi_fit)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic p.value
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)    9.09    0.0915       99.4       0
## 2 weight         0.241   0.00118     205.        0
```

$\hat{bmi} = 9.09 + 0.24 weight$

The coefficient of weight in this equation is 0.24, rounded to the hundredths, meaning that for every additional kilogram of weight, a person's BMI is expected to go up by approximately 0.24.

### Question 3

Now the coefficient of determination, or R-Squared, will be calculated for the linear model.

```
glance(bmi_fit)$r.squared
```

```
## [1] 0.8139379
```

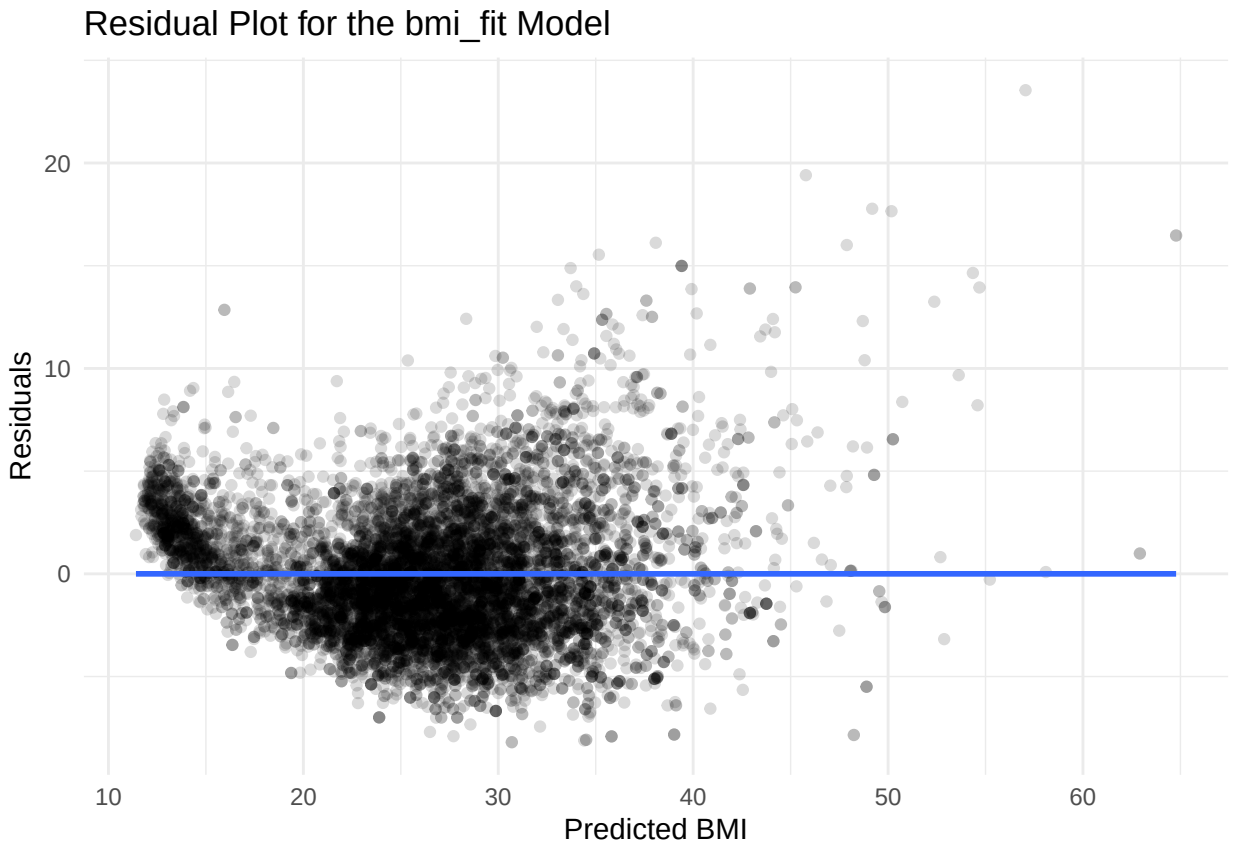
The coefficient of determination being approximately 0.81 tells us that 81% of the variability of the BMI can be explained by the weight from the linear model previously created. This high percentage suggests that the model created is a good fit for the data.

### Question 4

A residual plot will now be created to represent the linear model.

```
augment(bmi_fit) %>%
  ggplot(mapping = aes(x = .fitted, y = .resid)) +
  geom_point(alpha = 0.15) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x = "Predicted BMI",
       y = "Residuals",
       title = "Residual Plot for the bmi_fit Model") +
  theme_minimal()
```

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




Looking at this residual plot, it can be concluded that the linear model is not a great fit for the data. The residuals spanning across the plot are inconsistent in their range and shape, and it is known that seeing any

patterns on a residual plot is a sign that the model is not a great fit.

### Question 5

A new linear model will be created to predict systolic blood pressure as a function of weight, age, and gender of a person.

```
bp_fit <- lm(bp_sys_ave ~ weight + age + gender, data = nhanes)

tidy(bp_fit)
```

```
## # A tibble: 4 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)  92.3      0.580      159.   0.
## 2 weight        0.0937   0.00724     12.9  6.06e-38
## 3 age           0.414    0.00813     50.9  0.
## 4 gendermale    3.20     0.327        9.78 1.88e-22
```

The model equation for males: $\hat{bp} = 92.3 + 0.094weight + 0.414age + 3.20(1)$ The model equation for females: $\hat{bp} = 92.3 + 0.094weight + 0.414age + 3.20(0)$

As seen, the model equations for males and females differs only in that for the males, a "1" is substituted for the gender variable, meaning males here had a 3.20 higher systolic blood pressure on average than females.

```
predictbp <- tibble(age = 60, gender = "male", weight = 91)
predict(bp_fit, newdata = predictbp)
```

```
##        1
## 128.8575
```

With the code used after creating a new data frame with the desire values, the predicted systolic blood pressure of a 60-year old man weighing 200 pounds, or 91 kilograms, shows to be approximately 128.86 mm Hg.

### Question 6

Now a logistic model will be created to predict whether or not a patient has diabetes as a function of the patient's age, gender, and BMI.

```
diabetes_fit <- logistic_reg() %>%
  set_engine("glm") %>%
  fit(diabetes ~ age + gender + bmi, data = nhanes, family = "binomial")

tidy(diabetes_fit)
```

```
## # A tibble: 4 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)  -8.38     0.258      -32.5  5.16e-231
## 2 age           0.0582   0.00252     23.1  3.60e-118
## 3 gendermale    0.365    0.0833       4.38 1.21e-  5
## 4 bmi           0.0965   0.00560     17.2  1.25e- 66
```

This model's equation is: $\log(p/(1-p)) = -8.38 + 0.058age + 0.365gendermale + 0.097bmi$

### Question 7

Here a function will be written that outputs the inverse-logit of a number x.

```
inv_logit <- function(x) {
  exp(x)/(1 + exp(x))
}
```

This function could be useful if we wanted to find the probability of an event occurring, such as the probability of a patient having diabetes as a result of different factors from the logistic model from Question 6. In this case, after substituting the conditions we wanted to test for, meaning a certain age, gender, and BMI, the result of that model would be substituted for the x in the inverse-logit function to find the probability of the patient having diabetes.

### Question 8

We will now make use of the logistic regression model, its answer, and the inverse-logit to obtain the predicted probability of a 55-year old woman who has a BMI of 24 of having diabetes.

Substituting 55 for the age variable, 0 for the gender variable, and 24 for the BMI variable in the logistic regression model gives:

-2.862

Now substituting that result into the inverse logit function:

```
inv_logit(-2.862)
```

```
## [1] 0.05406433
```

The predicted probability of this patient having diabetes is approximately 0.054, or about 5.4% chance. This is a very small probability of this woman having diabetes, which is not entirely surprising because a BMI of 24 is not a very high one, and we might assume that people with a higher BMI have a higher chance of having diabetes. If we were to classify this result, we would classify the patient as not having diabetes.

### Question 9

Five outputs from the diabetes_fit model will now be mapped to the inv_logit function.

```
outputs <- list(c(-2.20, 0.01, -0.35, 1.15, 0.83))

map(outputs, inv_logit)
```

```
## [[1]]
## [1] 0.09975049 0.50249998 0.41338242 0.75951092 0.69635493
```

The result for the third patient is approximately 0.41, as seen from the mapping results. This means that given the patient's age, gender, and BMI, he or she has a predicted probability of 0.41, or about a 41% chance, of having diabetes.

### Question 10

In a binary classification situation with a high threshold, there needs to be a high probability of an event occurring for it to be classified as occurring. A high threshold may be desired if there were a situation where a risky drug or vaccine was to be tested on volunteers and there was a model that would predict who would have a high possibility of surviving the drug. A high threshold would be appropriate here because we would not want the model to predict a person to survive if there was only a 0.50 predicted probability of the person

surviving, for example. We would only want to test it on those who have a very high probability of surviving. A low threshold is wanted in situations where we want an event classified as occurring even if there is a relatively small chance of it occurring. Such a situation could be if the government had some sort of model to predict the probability of a person being a terrorist or a suspect in a crime that puts a whole country in danger. Here, even if the model predicts a probability of 0.10 of a person being a terrorist, we would want the government to investigate the person so as to make sure the person does not go free.

### Question 11

A site may not allow webscraping of its data if the data on the website is not accessible to everyone. If an account is necessary to get access to some data on a website, such as social media, then clearly the website does not want everyone to have access to certain data and would not allow webscraping for privacy reasons. Another scenario when a site may not allow webscraping is when it is a blog of some sort or a personal web page. In this case, the owner of the website probably would not want their personal information and comments to be scraped for another purpose.

### Question 12

This cartoon shows humans defeating robots in a robotic uprising because the robots chose to use inferior weapons, such as rocks and spears. This is because the algorithm the robots used to decide their weapons was based on battles that already occurred, and most battles ever fought were hundreds and thousands of years ago when technology was not so advanced. Therefore, the robots made a biased decision based on the data already available, which was hurtful to them. Similarly, the Amazon hiring process that was explained in the video also used an algorithm that rated candidates based on data already available from the past. The amount of male hires was already higher than woman hires in Amazon, so the algorithm used that information to make future decisions as well, which resulted in a biased decision by the algorithm.

### Question 13

A different file will now be read in and a few adjustments will be made as far as NA values, variable names and variable data types.

```
read_csv(file = "data/people.csv",
         na = c("N/A", "Unknown")) %>%
  janitor::clean_names() %>%
  mutate(height = fct_relevel(height, levels = c("short", "medium", "tall",
                                                 "very tall"))) %>%
  arrange(height)
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Name = col_character(),
##   `Date of Birth` = col_character(),
##   Height = col_character(),
##   `Number of Siblings` = col_double()
## )
```

```
## Warning: Problem with `mutate()` input `height`.
## i Outer names are only allowed for unnamed scalar atomic inputs
## i Input `height` is `fct_relevel(height, levels = c("short", "medium", "tall", "very tall"))`.
```

```
## Warning: Outer names are only allowed for unnamed scalar atomic inputs
```

```
## # A tibble: 6 x 4
##   name    date_of_birth height    number_of_siblings
##   <chr>   <chr>         <fct>                 <dbl>
## 1 Alice   9/9/1999      short                     3
## 2 Carlos  10/1/2002     medium                    1
## 3 Denise  1/28/1983     medium                   NA
## 4 Elaine  8/13/2006     tall                      2
## 5 Bob     2/4/1978      very tall                NA
## 6 Juanita 11/10/1965    <NA>                      0
```

Another way to ensure that columns had the appropriate data types would be to force R to consider them as a certain data type, using the col_types = and then listing from left to right the desired column data types. Code for arranging the factor levels was found here.