1 Final Project Introduction Statistics Fall 2020

Table of Contents

In [62]:

```
knitr::opts_chunk$set(echo = FALSE)
```

2 dataset: NHANES

The data we're going to work with comes from the National Health and Nutrition Examination Survey (NHANES) program at the CDC. You can read a lot more about NHANES on the CDC's website or Wikipedia. NHANES is a research program designed to assess the health and nutritional status of adults and children in the United States. The survey is one of the only to combine both survey questions and physical examinations. It began in the 1960s and since 1999 examines a nationally representative sample of about 5,000 people each year. The NHANES interview includes demographic, socioeconomic, dietary, and health-related questions. The physical exam includes medical, dental, and physiological measurements, as well as several standard laboratory tests. NHANES is used to determine the prevalence of major diseases and risk factors for those diseases. NHANES data are also the basis for national standards for measurements like height, weight, and blood pressure. Data from this survey is used in epidemiology studies and health sciences research, which help develop public health policy, direct and design health programs and services, and expand the health knowledge for the Nation.

We are using a small slice of this data. We're only using a handful of variables from the 2011-2012 survey years on about 5,000 individuals. The CDC uses a sampling strategy to purposefully oversample certain subpopulations like racial minorities. Naive analysis of the original NHANES data can lead to mistaken conclusions because the percentages of people from each racial group in the data are different from general population. The 5,000 individuals here are resampled from the larger NHANES study population to undo these oversampling effects, so you can treat this as if it were a simple random sample from the American population.

3 Part I - Descriptive Statistics

3.1 Load the data

In [63]:

```r
library(dplyr)
library(NHANES)
nh <- read.csv("nhanes.csv")
nh <- tbl_df(nh)
#display the variables of nhanes dataset
names(nh)
```

```
'ï..id'  'Gender'  'Age'  'Race'  'Education'  'MaritalStatus'  'RelationshipStatus'  'Insured'
'Income'  'Poverty'  'HomeRooms'  'HomeOwn'  'Work'  'Weight'  'Height'  'BMI'  'Pulse'
'BPSys'  'BPDia'  'Testosterone'  'HDLChol'  'TotChol'  'Diabetes'  'DiabetesAge'
'nPregnancies'  'nBabies'  'SleepHrsNight'  'PhysActive'  'PhysActiveDays'  'AlcoholDay'
'AlcoholYear'  'SmokingStatus'
```

`

3.2 Recall several built-in functions that are useful for working with data frames.

- Content:
    - head(): shows the first few rows
    - tail(): shows the last few rows
- Size:
    - dim(): returns a 2-element vector with the number of rows in the first element, and the number of columns as the second element (the dimensions of the object)

  - nrow(): returns the number of rows
  - ncol(): returns the number of columns
- Summary:
  - colnames() (or just names()): returns the column names
  - glimpse() (from dplyr): Returns a glimpse of your data, telling you the structure of the dataset and information about the class, length and content of each column

3.3 insert table of variables description in nhnanes dataset

In [64]:

```r
library(knitr)
nhanes.dd <- read.csv("nhanes_dd.csv")
kable(nhanes.dd)
```

```
|Variable           |Definition
|
|:------------------|:------------------------------------------------------
---------------------------------------------------------------------------
--------------------------------------------------------|
|id                 |A unique sample identifier
|
|Gender             |Gender (sex) of study participant coded as male or fem
ale
|
|Age                |Age in years at screening of study participant. Note:
Subjects 80 years or older were recorded as 80.
|
|Race               |Reported race of study participant, including non-Hisp
anic Asian category: Mexican, Hispanic, White, Black, Asian, or Other. Not
availale for 2009-10.                                   |
|Education          |Educational level of study participant Reported for pa
rticipants aged 20 years or older. One of 8thGrade, 9-11thGrade, HighSchoo
l, SomeCollege, or CollegeGrad.                          |
|MaritalStatus      |Marital status of study participant. Reported for part
icipants aged 20 years or older. One of Married, Widowed, Divorced, Separat
ed, NeverMarried, or LivePartner (living with partner). |
|RelationshipStatus |Simplification of MaritalStatus, coded as Committed if
MaritalStatus is Married or LivePartner, and Single otherwise.
|
|Insured            |Indicates whether the individual is covered by health
insurance.
|
|Income             |Numerical version of HHIncome derived from the middle
income in each category
|
|Poverty            |A ratio of family income to poverty guidelines. Smalle
r numbers indicate more poverty
|
|HomeRooms          |How many rooms are in home of study participant (count
ing kitchen but not bathroom). 13 rooms = 13 or more rooms.
|
|HomeOwn            |One of Home, Rent, or Other indicating whether the hom
e of study participant or someone in their family is owned, rented or occup
ied by some other arrangement.                          |
|Work               |Indicates whether the individual is current working or
not.
|
|Weight             |Weight in kg
|
|Height             |Standing height in cm. Reported for participants aged
2 years or older.
```

```
|
|BMI                |Body mass index (weight/height2 in kg/m2). Reported fo
r participants aged 2 years or older.
|
|Pulse              |60 second pulse rate
|
|BPSys              |Combined systolic blood pressure reading, following th
e procedure outlined for BPXSAR.
|
|BPDia              |Combined diastolic blood pressure reading, following t
he procedure outlined for BPXDAR.
|
|Testosterone       |Testerone total (ng/dL). Reported for participants age
d 6 years or older. Not available for 2009-2010.
|
|HDLChol            |Direct HDL cholesterol in mmol/L. Reported for partici
pants aged 6 years or older.
|
|TotChol            |Total HDL cholesterol in mmol/L. Reported for particip
ants aged 6 years or older.
|
|Diabetes           |Study participant told by a doctor or health professio
nal that they have diabetes. Reported for participants aged 1 year or older
as Yes or No.                                          |
|DiabetesAge        |Age of study participant when first told they had diab
etes. Reported for participants aged 1 year or older.
|
|nPregnancies       |How many times participant has been pregnant. Reported
for female participants aged 20 years or older.
|
|nBabies            |How many of participants deliveries resulted in live b
irths. Reported for female participants aged 20 years or older.
|
|SleepHrsNight      |Self-reported number of hours study participant usuall
y gets at night on weekdays or workdays. Reported for participants aged 16
years and older.                                        |
|PhysActive         |Participant does moderate or vigorous-intensity sport
s, fitness or recreational activities (Yes or No). Reported for participant
s 12 years or older.                                     |
|PhysActiveDays     |Number of days in a typical week that participant does
moderate or vigorous-intensity activity. Reported for participants 12 years
or older.                                             |
|AlcoholDay         |Average number of drinks consumed on days that partici
pant drank alcoholic beverages. Reported for participants aged 18 years or
older.                                                  |
|AlcoholYear        |Estimated number of days over the past year that parti
cipant drank alcoholic beverages. Reported for participants aged 18 years o
r older.                                                |
|SmokingStatus      |Smoking status: Current Former or Never.
|
```

In [65]:

```
# run the following commands
head(nh)
tail(nh)
dim(nh)
names(nh)
glimpse(nh)
```

A tibble: 6 × 32

ï..id	Gender	Age	Race	Education	MaritalStatus	RelationshipStatus	Insur
<int>	<chr>	<int>	<chr>	<chr>	<chr>	<chr>	<ch
62163	male	14	Asian	NA	NA	NA	
62172	female	43	Black	High School	NeverMarried	Single	
62174	male	80	White	College Grad	Married	Committed	
62174	male	80	White	College Grad	Married	Committed	
62175	male	5	White	NA	NA	NA	
62176	female	34	White	College Grad	Married	Committed	

A tibble: 6 × 32

ï..id	Gender	Age	Race	Education	MaritalStatus	RelationshipStatus	Insur
<int>	<chr>	<int>	<chr>	<chr>	<chr>	<chr>	<ch
71909	male	28	Mexican	9 - 11th Grade	NeverMarried	Single	
71909	male	28	Mexican	9 - 11th Grade	NeverMarried	Single	
71910	female	0	White	NA	NA	NA	
71911	male	27	Mexican	College Grad	Married	Committed	
71915	male	60	White	College Grad	NeverMarried	Single	
71915	male	60	White	College Grad	NeverMarried	Single	

```
5000   32
```

```
'ï..id'  'Gender'  'Age'  'Race'  'Education'  'MaritalStatus'  'RelationshipStatus'
'Insured'  'Income'  'Poverty'  'HomeRooms'  'HomeOwn'  'Work'  'Weight'  'Height'
'BMI'  'Pulse'  'BPSys'  'BPDia'  'Testosterone'  'HDLChol'  'TotChol'  'Diabetes'
```

```
'DiabetesAge'  'nPregnancies'  'nBabies'  'SleepHrsNight'  'PhysActive'
'PhysActiveDays'  'AlcoholDay'  'AlcoholYear'  'SmokingStatus'
```

```
Rows: 5,000
Columns: 32
$ ï..id              <int> 62163, 62172, 62174, 62174, 62175, 62176, 6217
8,...
$ Gender             <chr> "male", "female", "male", "male", "male", "fema
l...
$ Age                <int> 14, 43, 80, 80, 5, 34, 80, 35, 17, 15, 57, 57,
 5...
$ Race               <chr> "Asian", "Black", "White", "White", "White", "W
h...
$ Education          <chr> NA, "High School", "College Grad", "College Gra
d...
$ MaritalStatus      <chr> NA, "NeverMarried", "Married", "Married", NA,
 "M...
$ RelationshipStatus <chr> NA, "Single", "Committed", "Committed", NA, "Co
m...
$ Insured            <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Ye
s",...
$ Income             <int> 100000, 22500, 70000, 70000, 12500, 100000, 250
0...
$ Poverty            <dbl> 4.07, 2.02, 4.30, 4.30, 0.39, 5.00, 0.05, 0.87,
 ...
$ HomeRooms          <int> 6, 4, 7, 7, 7, 8, 6, 6, 6, 4, 4, 4, 4, 4, 12, 1
2...
$ HomeOwn            <chr> "Rent", "Rent", "Own", "Own", "Rent", "Own", "O
w...
$ Work               <chr> NA, "NotWorking", "NotWorking", "NotWorking", N
A...
$ Weight             <dbl> 49.4, 98.6, 95.8, 95.8, 23.9, 68.7, 85.9, 89.0,
 ...
$ Height             <dbl> 168.9, 172.0, 168.1, 168.1, 119.8, 171.6, 173.
5,...
$ BMI                <dbl> 17.3, 33.3, 33.9, 33.9, 16.7, 23.3, 28.5, 27.9,
 ...
$ Pulse              <int> 72, 80, 56, 56, NA, 92, 68, 66, 86, 76, 84, 84,
 ...
$ BPSys              <int> 107, 103, 97, 97, NA, 107, 121, 107, 108, 113,
 1...
$ BPDia              <int> 37, 72, 39, 39, NA, 69, 72, 66, 64, 27, 65, 65,
 ...
$ Testosterone       <dbl> 274.95, 47.53, 642.82, 642.82, NA, 21.11, 562.7
8...
$ HDLChol            <dbl> 1.14, 1.89, 1.40, 1.40, NA, 1.42, 1.22, 0.85,
 1....
$ TotChol            <dbl> 3.98, 4.37, 5.25, 5.25, NA, 4.42, 5.20, 3.70,
 3....
$ Diabetes           <chr> "No", "No", "No", "No", "No", "No", "No", "No",
 ...
$ DiabetesAge        <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
 ...
$ nPregnancies       <int> NA, 3, NA, NA, NA, 5, NA, NA, NA, NA, NA, NA, N
A...
$ nBabies            <int> NA, 2, NA, NA, NA, 2, NA, NA, NA, NA, NA, NA, N
A...
```

```
$ SleepHrsNight      <int> NA, 8, 9, 9, NA, 7, 6, 7, 7, NA, 8, 8, 8, 8, 6,
 ...
$ PhysActive         <chr> "No", "No", "No", "No", NA, "Yes", "No", "No",
 "...
$ PhysActiveDays     <int> 1, 2, 7, 5, 7, 5, NA, NA, 4, 7, 2, NA, 7, NA, N
A...
$ AlcoholDay         <int> NA, 3, NA, NA, NA, 2, NA, 1, NA, NA, 1, 1, 1,
 1,...
$ AlcoholYear        <int> NA, 104, 0, 0, NA, 104, NA, 2, NA, NA, 260, 26
0,...
$ SmokingStatus      <chr> NA, "Current", "Never", "Never", NA, "Never",
 "N...
```

3.4 Descriptive statistics

We can access individual variables within a data frame using the $operator, e.g., mydataframe$specificVariable. Let’s print out all the Race values in the data. Let’s then see what are the unique values of each. Then let’s calculate the mean, median, and range of the Age variable.

If you run the `summary()` function on a data frame, you get some very basic summary statistics on each variable in the data.

Exercise 1 a) display race values b) get unique values of Race c) length of Race d) Read the functions that dplyr (https://dplyr.tidyverse.org/) supports e) do the d) using dplyr way

3.5 Missing data

Let's try taking the mean of a `income` variable.

In [66]:

```
mean(nh$Income)
```

<NA>

What happened there? NA indicates missing data. Take a look at the Income variable.

In [67]:

```
glimpse(nh$Income)
```

```
 int [1:5000] 100000 22500 70000 70000 12500 100000 2500 22500 22500 30000
...
```

Notice that there are lots of missing values for Income. Trying to get the mean a bunch of observations with some missing data returns a missing value by default. This is almost universally the case with all summary statistics – a single NA will cause the summary to return NA. Now look at the help for ?mean. Notice the na.rm argument. This is a logical (i.e., TRUE or FALSE) value indicating whether or not missing values should be removed prior to computing the mean. By default, it’s set to FALSE. Now try it again.

In [68]:

```
mean(nh$Income, na.rm=TRUE)
```

```
57077.6552022496
```

The `is.na()` function tells you if a value is missing. Get the `sum( )`` of that vector, which adds up all the TRUEs to tell you how many of the values are missing.

In [69]:

```
#is.na(nh$Income))
sum(is.na(nh$Income))
```

```
377
```

R is.na Function Example (remove, replace, count, if else, is not NA)

Before we can start, let’s create some example data in R (or R Studio).

In [70]:

```
set.seed(11991)                               # Set seed
N <- 1000                                     # Sample size

x_num <- round(rnorm(N, 0, 5))                # Numeric
x_fac <- as.factor(round(runif(N, 0, 3)))     # Factor
x_cha <- sample(letters, N, replace = TRUE)   # Character

x_num[rbinom(N, 1, 0.2) == 1] <- NA           # 20% missings
x_fac[rbinom(N, 1, 0.3) == 1] <- NA           # 30% missings
x_cha[rbinom(N, 1, 0.05) == 1] <- NA          # 5% missings

data <- data.frame(x_num, x_fac, x_cha,       # Create data frame
                   stringsAsFactors = FALSE)
head(data)                                    # First rows of data
```

A data.frame: 6 × 3

	x_num	x_fac	x_cha
	<dbl>	<fct>	<chr>
1	8	2	p
2	0	NA	a
3	-4	2	j
4	NA	1	x
5	-6	1	s
6	-3	NA	k

Our data consists of three columns, each of them with a different class: numeric, factor, and character. This is how the first six lines of our data look like:

Let’s apply the is.na function to our **whole data set**:

In [71]:

```
head(is.na(data))
```

A matrix: 6 × 3 of type lgl

x_num	x_fac	x_cha
FALSE	FALSE	FALSE
FALSE	TRUE	FALSE
FALSE	FALSE	FALSE
TRUE	FALSE	FALSE
FALSE	FALSE	FALSE
FALSE	TRUE	FALSE

We are also able to check whether there is or is not an NA value in a column or vector:

In [72]:

```
head(is.na(data$x_num))  # Works for numeric ...
head(is.na(data$x_fac))  # ... factor ...
head(is.na(data$x_cha))  # ... and character

head(!is.na(data$x_num)) # The explanation mark still works
head(!is.na(data$x_fac))
head(!is.na(data$x_cha))
```

```
FALSE  FALSE  FALSE  TRUE  FALSE  FALSE

FALSE  TRUE  FALSE  FALSE  FALSE  TRUE

FALSE  FALSE  FALSE  FALSE  FALSE  FALSE

TRUE  TRUE  TRUE  FALSE  TRUE  TRUE

TRUE  FALSE  TRUE  TRUE  TRUE  FALSE

TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

That’s nice, but the real power of `is.na` becomes visible in combination with other functions — And that’s exactly what I’m going to show you now.

In [73]:

```
# is.na in Combination with Other R Functions
#  Remove NAs of Vector or Column
length(data$x_num)
is.na_remove <- data$x_num[!is.na(data$x_num)]

length(is.na_remove)

### Replace NAs with Other Values (i.e. mean)


is.na_replace_mean <- data$x_num                        # Duplicate fi
x_num_mean <- mean(is.na_replace_mean, na.rm = TRUE)    # Calculate me
is.na_replace_mean[is.na(is.na_replace_mean)] <- x_num_mean # Replace by m


#In case of characters or factors, it is also possible in R to set NA to b


is.na_blank_cha <- data$x_cha                           # Duplicate ch
is.na_blank_cha[is.na(is.na_blank_cha)] <- ""           # Class charac

is.na_blank_fac <- data$x_fac                           # Duplicate fa
is.na_blank_fac <- as.character(is.na_blank_fac)        # Convert temp
is.na_blank_fac[is.na(is.na_blank_fac)] <- ""           # Class charac
is.na_blank_fac <- as.factor(is.na_blank_fac)           # Recode back
```

```
1000

799
```

3.6 Count NAs via sum & colSums

Combined with the R function sum, we can count the amount of NAs in our columns. According to our previous data generation, it should be approximately 20% in x_num, 30% in x_fac, and 5% in x_cha.

In [74]:

```
sum(is.na(data$x_num)) # 213 missings in the first column
sum(is.na(data$x_fac)) # 322 missings in the second column
sum(is.na(data$x_cha)) # 47 missings in the third column


# If we want to count NAs in multiple columns at the same time, we can use

colSums(is.na(data))



# Detect if there are any NAs

# We can also test, if there is at least 1 missing value in a column of ou


any(is.na(data$x_num))
```

```
201

305

54

x_num
201
x_fac
305
x_cha
54


TRUE
```

3.6.1 Locate NAs via which

In combination with the which function, is.na can be used to identify the positioning of NAs:

In [75]:

```
head(which(is.na(data$x_num)))
```

```
4  10  13  15  17  18
```

Our first column has missing values at the positions 4, 5, 14, 17, 22, 23 and so forth.

3.6.2 if & if else

Missing values have to be considered in our programming routines, e.g. within the if statement or within for loops.

In the following example, I’m printing “Damn, it’s NA” to the R Studio console whenever a missing occurs; and “Wow, that’s awesome” in case of an observed value.

In [76]:

```
for(i in 1:10) {
  if(is.na(data$x_num[i])) {
    print("Damn, it's NA")
  }
  else {
    print("Wow, that's awesome")
  }
}
```

```
[1] "Wow, that's awesome"
[1] "Wow, that's awesome"
[1] "Wow, that's awesome"
[1] "Damn, it's NA"
[1] "Wow, that's awesome"
[1] "Wow, that's awesome"
[1] "Wow, that's awesome"
[1] "Wow, that's awesome"
[1] "Wow, that's awesome"
[1] "Damn, it's NA"
```

Note: Within the if statement we use is na instead of equal to — the approach we would usually use in case of observed values (e.g. if(x[i] == 5)).

Even easier to apply: the ifelse function.

In [77]:

```
head(ifelse(is.na(data$x_num), "Damn, it's NA", "Wow, that's awesome"))
```

'Wow, that\'s awesome'   'Wow, that\'s awesome'   'Wow, that\'s awesome'   'Damn, it\'s NA'
'Wow, that\'s awesome'   'Wow, that\'s awesome'

There are a few handy functions in the `Tmisc` package for summarizing missingness in a data frame. You can graphically display missingness in a data frame as holes on a black canvas with `gg_na()` (use ggplot2 to plot NA values), or show a table of all the variables and the missingness level with propmiss().

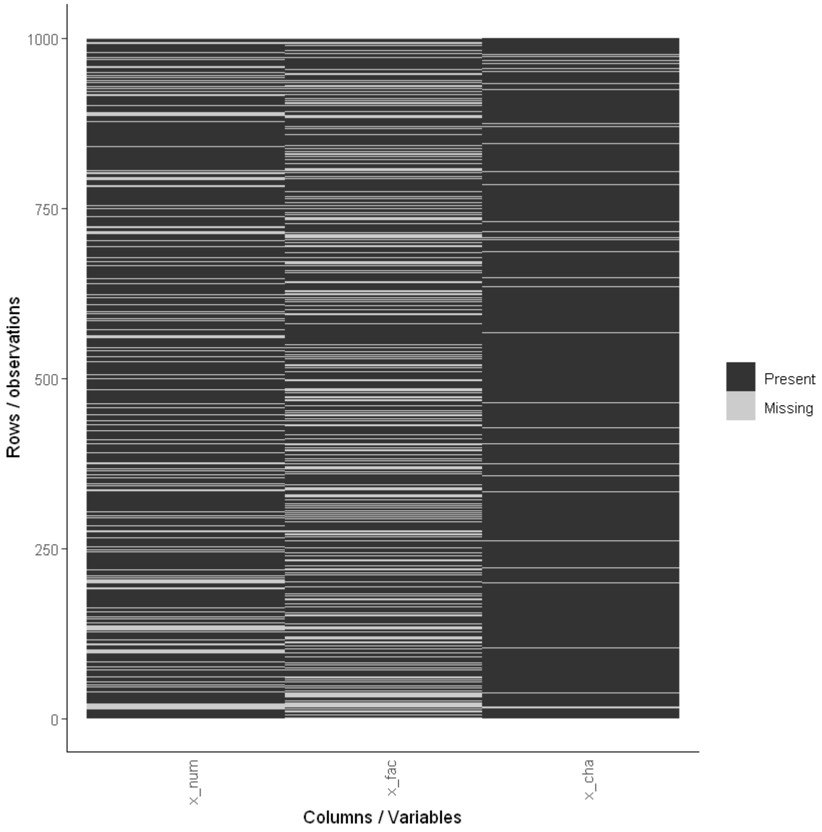
In [78]:

```
# Install if you don't have it already
# install.packages("Tmisc")

# Load Tmisc
library(Tmisc)
gg_na(data)


propmiss(data)
```

```
Warning message:
"'propmiss' is deprecated.
Use 'Use summarize(across(everything(), ~sum(is.na(.))/n())' instead.
See help("Deprecated")"
```

A tibble: 3 × 4

var	nmiss	n	propmiss
<chr>	<dbl>	<dbl>	<dbl>
x_num	201	1000	0.201
x_fac	305	1000	0.305
x_cha	54	1000	0.054



Exercise 2 Apply the above functions to other column of the dataset

4 Part II: Explatory Data Analysis (EDA)

It’s always worth examining your data visually before you start any statistical analysis or hypothesis testing. The big ones are histograms and scatterplots.

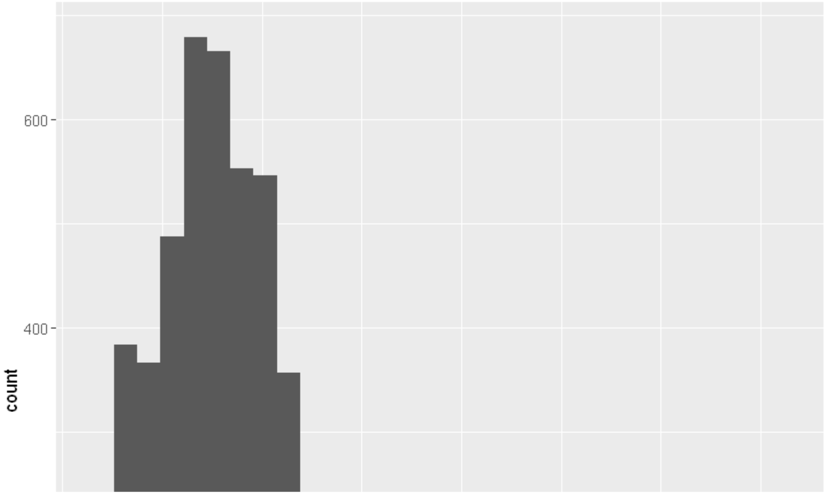
4.1 Histograms

We can learn a lot from the data just looking at the value distributions of particular variables. Let’s make some histograms with ggplot2. Looking at BMI shows a few extreme outliers. Looking at weight initially shows us that the units are probably in kg. Replotting that in lbs with more bins shows a clear bimodal distribution. Are there kids in this data? The age distribution shows us the answer is yes.

In [79]:

```
library(ggplot2)
ggplot(nh, aes(BMI)) + geom_histogram(bins=30)


ggplot(nh, aes(Weight)) + geom_histogram(bins=30)

# In pounds, more bins
ggplot(nh, aes(Weight*2.2)) + geom_histogram(bins=80)


ggplot(nh, aes(Age)) + geom_histogram(bins=30)
```

```
Warning message:
"Removed 166 rows containing non-finite values (stat_bin)."
Warning message:
"Removed 31 rows containing non-finite values (stat_bin)."
```

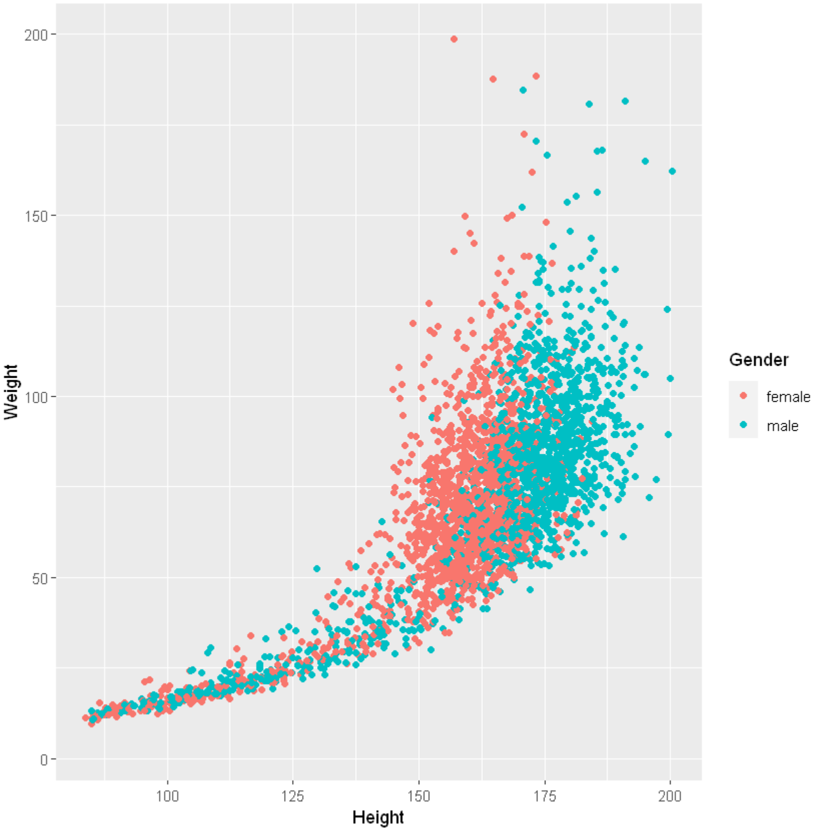


4.2 Scatterplots

Let’s look at how a few different variables relate to each other. E.g., height and weight:

```
In [80]:  ggplot(nh, aes(Height, Weight, col=Gender)) + geom_point()
```

```
Warning message:
"Removed 166 rows containing missing values (geom_point)."
```

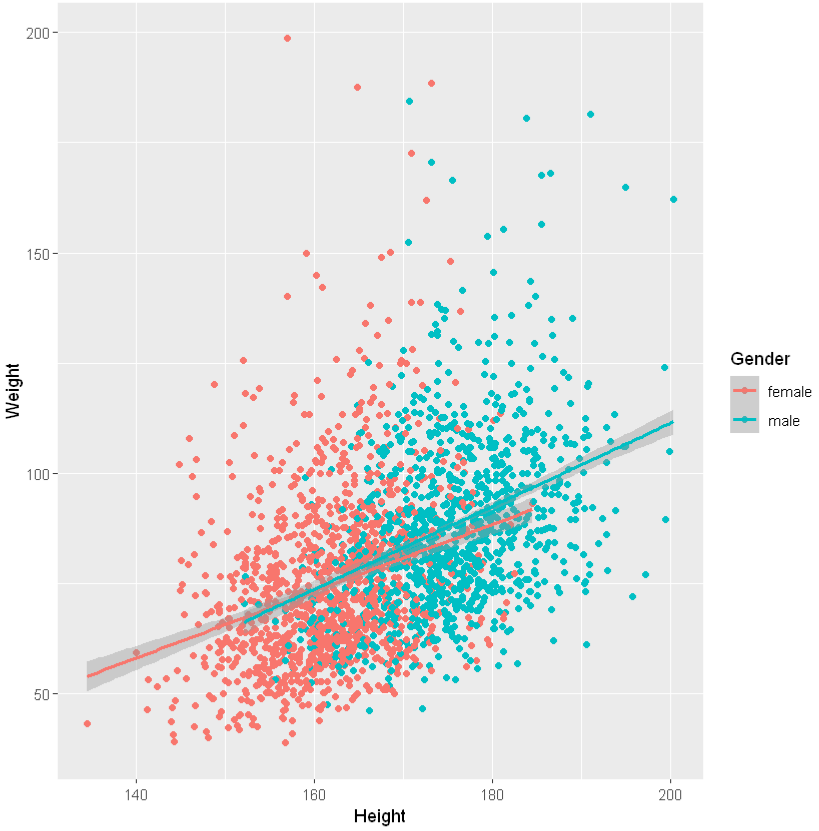


Let’s filter out all the kids, draw trend lines using a linear model:

In [81]:

```
nh %>%
  filter(Age>=18) %>%
  ggplot(aes(Height, Weight, col=Gender)) +
    geom_point() +
    geom_smooth(method="lm")
```

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

Warning message:
"Removed 31 rows containing non-finite values (stat_smooth)."
Warning message:
"Removed 31 rows containing missing values (geom_point)."
```



Exercise 3 a) Make some histograms with ggplot2 of 2 variables. b) Look at BMI and indicate whether there outliers. c) Look at weight. What their distribution looks like? d) Check the age distribution. Are there kids in this data? Explain

Exercise 4

1. What’s the mean 60-second pulse rate for all participants in the data?
2. What’s the range of values for diastolic blood pressure in all participants? (Hint: see help for min(), max(), and range() functions, e.g., enter ?range without the parentheses to get help).
3. What are the median, lower, and upper quartiles for the age of all participants? (Hint: see help for median, or better yet, quantile).
4. What’s the variance and standard deviation for income among all participants?

5 Continuous variables

5.1 T-tests

First let’s create a new dataset from nh called nha that only has adults. To prevent us from making any mistakes downstream, let’s remove the nh object.

In [82]:

```
nha <- filter(nh, Age>=18)
rm(nh)
```

Let’s do a few two-sample t-tests to test for differences in means between two groups. The function for a t-test is `t.test()` . See the help for `?t.test` . We’ll be using the formula method. The usage is `t.test(response~group, data=myDataFrame)` .

Exercise 5

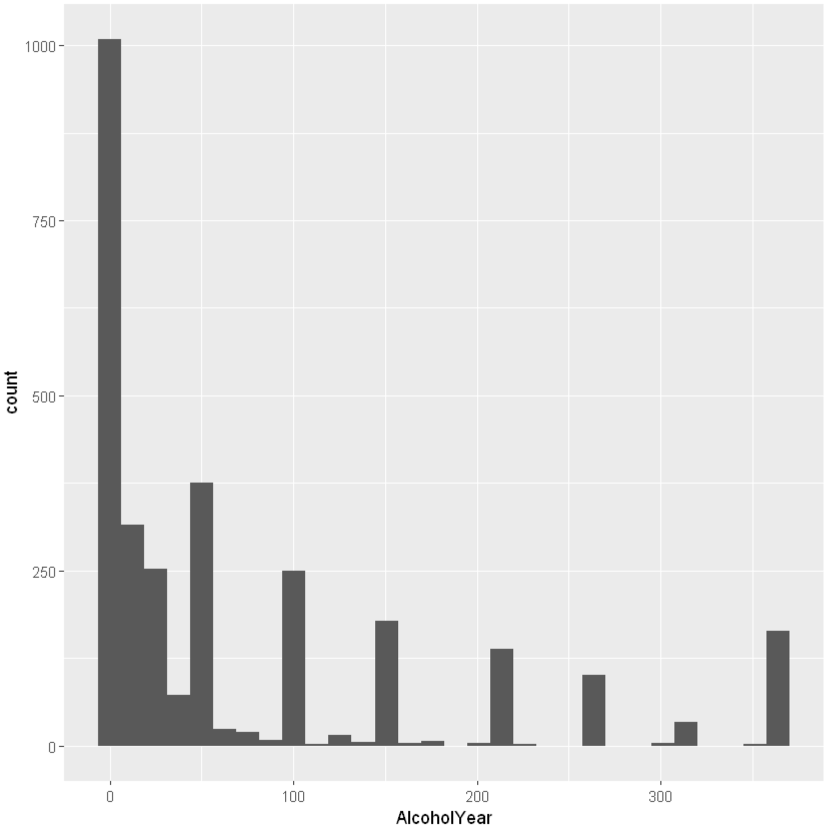
1. Are there differences in age for males versus females in this dataset?
2. Does BMI differ between diabetics and non-diabetics?
3. Do single or married/cohabitating people drink more alcohol? Is this relationship significant?

5.2 Wilcoxon test

Another assumption of the t-test is that data is normally distributed. Looking at the histogram for AlcoholYear shows that this data clearly isn’t.

```
In [83]:  ggplot(nha, aes(AlcoholYear)) + geom_histogram()
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

Warning message:
"Removed 723 rows containing non-finite values (stat_bin)."
```



The Wilcoxon rank-sum test (a.k.a. Mann-Whitney U test) is a nonparametric test of differences in mean that does not require normally distributed data. When data is perfectly normal, the t-test is uniformly more powerful. But when this assumption is violated, the t-test is unreliable. This test is called in a similar way as the t-test.

In [84]:

```
wilcox.test(AlcoholYear~RelationshipStatus, data=nha)
```

```
	Wilcoxon rank sum test with continuity correction

data:  AlcoholYear by RelationshipStatus
W = 1067955, p-value = 0.0001659
alternative hypothesis: true location shift is not equal to 0
```

5.3 ANOVA

Where t-tests and their nonparametric substitutes are used for assessing the differences in means between two groups, ANOVA is used to assess the significance of differences in means between multiple groups. In fact, a t-test is just a specific case of ANOVA when you only have two groups. And both t-tests and ANOVA are just specific cases of linear regression, where you’re trying to fit a model describing how a continuous outcome (e.g., BMI) changes with some predictor variable (e.g., diabetic status, race, age, etc.). The distinction is largely semantic – with a linear model you’re asking, “do levels of a categorical variable affect the response?” where with ANOVA or t-tests you’re asking, “does the mean response differ between levels of a categorical variable?”

Let’s examine the relationship between BMI and relationship status (RelationshipStatus was derived from MaritalStatus, coded as Committed if MaritalStatus is Married or LivePartner, and Single otherwise). Let’s first do this with a t-test, and for now, let’s assume that the variances between groups are equal.

In [85]:

```
t.test(BMI~RelationshipStatus, data=nha, var.equal=TRUE)
```

```
	Two Sample t-test

data:  BMI by RelationshipStatus
t = -1.5319, df = 3552, p-value = 0.1256
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.77817842  0.09552936
sample estimates:
mean in group Committed    mean in group Single 
               28.51343                28.85475 
```

It looks like single people have a very slightly higher BMI than those in a committed relationship, but the magnitude of the difference is trivial, and the difference is not significant. Now, let’s do the same test in a linear modeling framework. First, let’s create the fitted model and store it in an object called fit.

In [86]:

```
fit <- lm(BMI~RelationshipStatus, data=nha)
```

You can display the object itself, but that isn't too interesting. You can get the more familiar ANOVA table by calling the anova() function on the fit object. More generally, the summary() function on a linear model object will tell you much more.

In [87]:

```
anova(fit)
```

A anova: 2 × 5

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
RelationshipStatus	1	98.31983	98.31983	2.346685	0.1256388
Residuals	3552	148819.30437	41.89733	NA	NA

5.3.1 ANOVA

Recap: t-tests are for assessing the differences in means between two groups. A t-test is a specific case of ANOVA, which is a specific case of a linear model. Let's run ANOVA, but this time looking for differences in means between more than two groups.

Let's look at the relationship between smoking status (Never, Former, or Current), and BMI.

In [88]:

```
fit <- lm(BMI~SmokingStatus, data=nha)
anova(fit)
```

A anova: 2 × 5

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
SmokingStatus	2	1411.01	705.50494	16.98847	4.539974e-08
Residuals	3553	147550.57	41.52845	NA	NA

The ANOVA table tells us that there is a significant difference in means between current, former, and never smokers (p=4.54×10−8)

6 Linear regression

Linear models seek to explain the relationship between a variable of interest, our Y, outcome, response, or dependent variable, and one or more X, predictor, or independent variables. Previously we talked about t-tests or ANOVA in the context of a simple linear regression model with only a single predictor variable, X :

$$Y = \beta_0 + \beta_1 X \tag{1}$$

But you can have multiple predictors in a linear model that are all additive, accounting for the effects of the others:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon \tag{2}$$

- Y is the response
- X_1 and X_2 are the predictors
- β_0 is the intercept, and β_1, β_2 etc are coefficients that describe what 1 -unit changes in X_1 and X_2 do to the outcome variable Y.
- ϵ is random error. Our model will not perfectly predict Y. It will be off by some random amount. We assume this amount is a random draw from a Normal distribution with mean 0 and standard deviation σ.

Building a linear model means we propose a linear model and then estimate the coefficients and the variance of the error term. Above, this means estimating $\beta_0, \beta_1, \beta_2$ and σ. This is what we do in R Let's look at the relationship between height and weight.

In [89]:

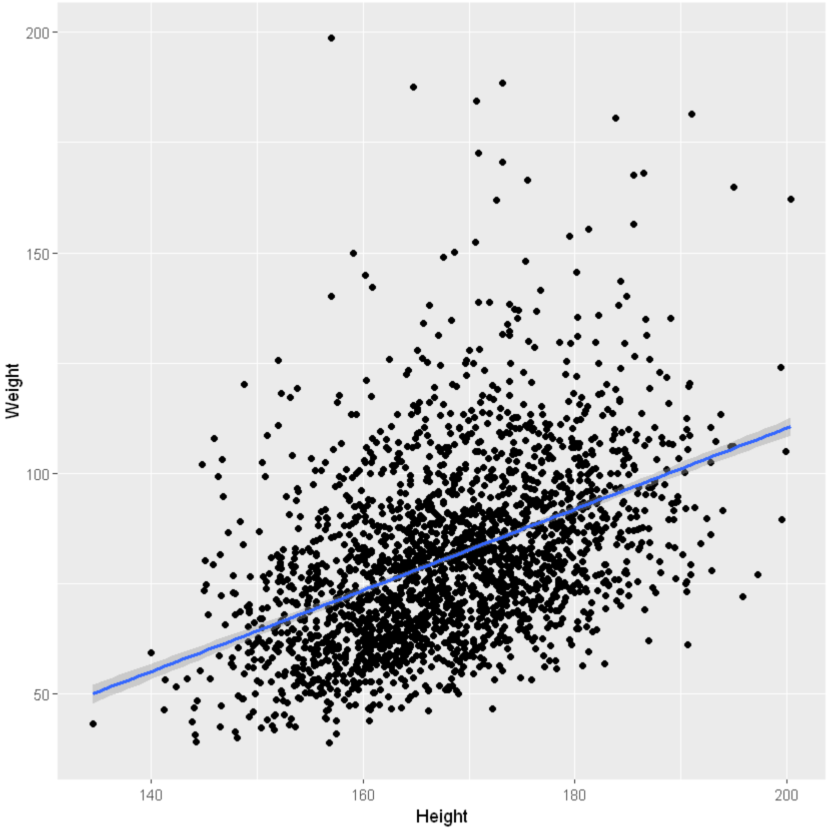
```
fit <- lm(Weight~Height, data=nha)
summary(fit)
```

```

Call:
lm(formula = Weight ~ Height, data = nha)

Residuals:
    Min      1Q  Median      3Q     Max 
-40.339 -13.109  -2.658   9.309 127.972 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -73.70590    5.08110  -14.51   <2e-16 ***
Height        0.91996    0.03003   30.63   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 18.57 on 3674 degrees of freedom
  (31 observations deleted due to missingness)
Multiple R-squared:  0.2034,	Adjusted R-squared:  0.2032 
F-statistic: 938.4 on 1 and 3674 DF,  p-value: < 2.2e-16
```

The relationship is highly significant $\left(\mathrm{P} < 2.2 \times 10^{-16}\right)$. The intercept term is not very useful most of the time. Here it shows us what the value of Weight would be when Height=0, which could never happen. The Height coefficient is meaningful - each one unit increase in height results in a 0.92 increase in the corresponding unit of weight. Let's visualize that relationship:

```
In [90]:  ggplot(nha, aes(x=Height, y=Weight)) + geom_point() + geom_smooth(method="
```

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

Warning message:
"Removed 31 rows containing non-finite values (stat_smooth)."
Warning message:
"Removed 31 rows containing missing values (geom_point)."
```

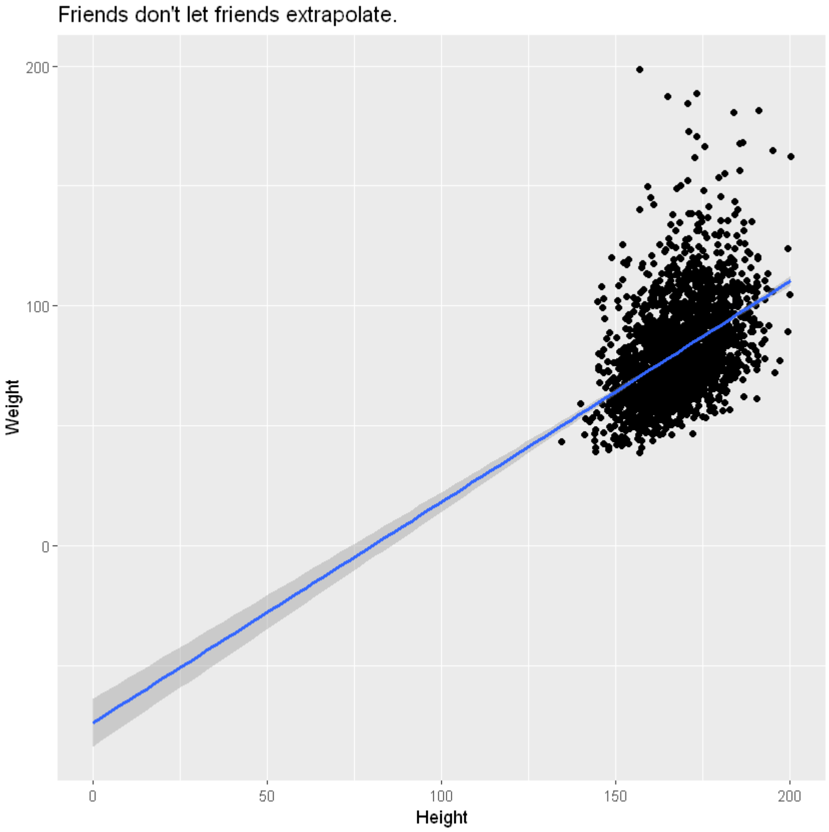


By default, this is only going to show the prediction over the range of the data. This is important! You never want to try to extrapolate response variables outside of the range of your predictor(s). For example, the linear model tells us that weight is -73.7kg when height is zero. We can extend the predicted model / regression line past the lowest value of the data down to height=0. The bands on the confidence interval tell us that the model is apparently confident within the regions defined by the gray boundary. But this is silly – we would never see a height of zero, and predicting past the range of the available training data is never a good idea.

In [91]:

```
ggplot(nha, aes(x=Height, y=Weight)) +
  geom_point() +
  geom_smooth(method="lm", fullrange=TRUE) +
  xlim(0, NA) +
  ggtitle("Friends don't let friends extrapolate.")
```

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

Warning message:
"Removed 31 rows containing non-finite values (stat_smooth)."
Warning message:
"Removed 31 rows containing missing values (geom_point)."
```



7 Multiple regression

Finally, let’s do a multiple linear regression analysis, where we attempt to model the effect of multiple predictor variables at once on some outcome. First, let’s look at the effect of physical activity on testosterone levels. Let’s do this with a t-test and linear regression, showing that you get the same results.

In [92]:

```
t.test(Testosterone~PhysActive, data=nha)
```

```
	Welch Two Sample t-test

data:  Testosterone by PhysActive
t = -2.4349, df = 3335.2, p-value = 0.01495
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -34.781568  -3.752469
sample estimates:
 mean in group No mean in group Yes 
         207.5645          226.8315 
```

In [93]:

```
summary(lm(Testosterone~PhysActive, data=nha))
```

```
Call:
lm(formula = Testosterone ~ PhysActive, data = nha)

Residuals:
   Min     1Q Median     3Q    Max 
-224.5 -196.5 -115.9  167.0 1588.0 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)    207.565      5.873   35.34   <2e-16 ***
PhysActiveYes   19.267      7.929    2.43   0.0152 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 231.4 on 3436 degrees of freedom
  (269 observations deleted due to missingness)
Multiple R-squared:  0.001715,	Adjusted R-squared:  0.001425 
F-statistic: 5.904 on 1 and 3436 DF,  p-value: 0.01516
```

In both cases, the p-value is significant (p=0.01516), and the result suggest that increased physical activity is associated with increased testosterone levels. Does increasing your physical activity increase your testosterone levels? Or is it the other way – will increased testosterone encourage more physical activity? Or is it none of the above – is the apparent relationship between physical activity and testosterone levels only apparent because both are correlated with yet a third, unaccounted for variable? Let’s throw Age into the model as well.

In [94]:

```
summary(lm(Testosterone~PhysActive+Age, data=nha))
```

```
Call:
lm(formula = Testosterone ~ PhysActive + Age, data = nha)

Residuals:
   Min     1Q Median     3Q    Max 
-238.6 -196.8 -112.3  167.4 1598.1 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)   247.8828    13.0853  18.944  < 2e-16 ***
PhysActiveYes  13.6740     8.0815   1.692 0.090735 .  
Age            -0.8003     0.2322  -3.447 0.000574 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 231 on 3435 degrees of freedom
  (269 observations deleted due to missingness)
Multiple R-squared:  0.005156,	Adjusted R-squared:  0.004577 
F-statistic: 8.901 on 2 and 3435 DF,  p-value: 0.0001394
```

This shows us that after accounting for age that the testosterone / physical activity link is no longer significant. Every 1-year increase in age results in a highly significant decrease in testosterone, and since increasing age is also likely associated with decreased physical activity, perhaps age is the confounder that makes this relationship apparent.

Adding other predictors can also swing things the other way. We know that men have much higher testosterone levels than females. Sex is probably the single best predictor of testosterone levels in our dataset. By not accounting for this effect, our unaccounted-for variation remains very high. By accounting for Gender, we now reduce the residual error in the model, and the physical activity effect once again becomes significant. Also notice that our model fits much better (higher R-squared), and is much more significant overall.

In [95]:

```
summary(lm(Testosterone~PhysActive+Age+Gender, data=nha))
```

```
Call:
lm(formula = Testosterone ~ PhysActive + Age + Gender, data = nha)

Residuals:
    Min      1Q  Median      3Q     Max 
-397.91  -31.01   -4.42   20.50 1400.90 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)    46.6931     7.5729   6.166 7.83e-10 ***
PhysActiveYes   9.2749     4.4617   2.079   0.0377 *  
Age            -0.5904     0.1282  -4.605 4.28e-06 ***
Gendermale    385.1989     4.3512  88.526  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 127.5 on 3434 degrees of freedom
  (269 observations deleted due to missingness)
Multiple R-squared:  0.6969,	Adjusted R-squared:  0.6966 
F-statistic:  2632 on 3 and 3434 DF,  p-value: < 2.2e-16
```

We've only looked at the `summary ()` and `anova ()` functions for extracting information from an lm class object. There are several other accessor functions that can be used on a linear model object. Check out the help page for each one of these to learn more.

- `coefficients ( )`
- `predict.lm ()`
- `fitted.values ()`
- `residuals ()`

Exercise 6 Is the average BMI different in single people versus those in a committed relationship? Perform a t-test.

Exercise 7

2. The work variable is coded "Looking" (n=159), “NotWorking" (n=1317), and "Working" $(n = 2230)$

    - Fit a linear model. Assign this to an object called fit. What does the fit object tell you when you display it directly?
    - Run an anova () to get the ANOVA table. Is the model significant?
    - Instead of thinking of this as ANOVA, think of it as a linear model. After you've thought about it, get some summary () statistics on the fit. Do these results agree with the ANOVA model?

Exercise 8 Examine the relationship between HDL cholesterol levels ($\mathrm{HDLChol}$) and whether someone has diabetes or not (Diabetes).

```
- Is there a difference in means between diabetics and nondiabetics? P
erform a t-test without a Welch correction (that is, assuming equal va
riances - see ?t.test for help).

- Do the same analysis in a linear modeling framework.

- Does the relationship hold when adjusting for weight?

- What about when adjusting for weight, Age, Gender, PhysActive (wheth
ersomeone participates in moderate or vigorous-intensity sports, fitne
ss or recreational activities, coded as yes/no). What is the effect of
each of these explanatory variables?
```

8 Discrete variables

Until now we’ve only discussed analyzing continuous outcomes / dependent variables. We’ve tested for differences in means between two groups with t-tests, differences among means between n groups with ANOVA, and more general relationships using linear regression. In all of these cases, the dependent variable, i.e., the outcome, or Y variable, was continuous, and usually normally distributed. What if our outcome variable is discrete, e.g., “Yes/No”, “Mutant/WT”, “Case/Control”, etc.? Here we use a different set of procedures for assessing significant associations.

9 Contingency tables

In statistics, a contingency table (also known as a cross tabulation or crosstab) is a type of table in a matrix format that displays the (multivariate) frequency distribution of the variables. They are heavily used in survey research, business intelligence, engineering, and scientific research. They provide a basic picture of the interrelation between two variables and can help find interactions between them.

The xtabs() function is useful for creating contingency tables from categorical variables. Let’s create a gender by diabetes status contingency table, and assign it to an object called xt. After making the assignment, type the name of the object to view it.

In [96]:

```
xt <- xtabs(~Gender+Diabetes, data=nha)
xt
```

```
        Diabetes
Gender     No  Yes
  female 1692  164
  male   1653  198
```

There are two useful functions, `addmargins()` and `prop.table()` that add more information or manipulate how the data is displayed. By default, `prop.table()` will divide the number of observations in each cell by the total. But you may want to specify which margin you want to get proportions over. Let's do this for the first (row) margin.

In [97]:

```
# Add marginal totals
addmargins(xt)
```

A table: 3 × 3 of type dbl

	No	Yes	Sum
female	1692	164	1856
male	1653	198	1851
Sum	3345	362	3707

In [98]:

```
# Get the proportional table
prop.table(xt)
```

```
        Diabetes
Gender           No        Yes
  female 0.45643377 0.04424063
  male   0.44591314 0.05341246
```

In [99]:

```
# That wasn't really what we wanted.
# Do this over the first (row) margin only.
prop.table(xt, margin=1)
```

```
        Diabetes
Gender           No        Yes
  female 0.91163793 0.08836207
  male   0.89303079 0.10696921
```

Looks like men have slightly higher rates of diabetes than women. But is this significant?

The chi-square test is used to assess the independence of these two factors. That is, if the null hypothesis that gender and diabetes are independent is true, the we would expect a proportionally equal number of diabetics across each sex. Males seem to be at slightly higher risk than females, but the difference is just short of statistically significant.

In [100]:

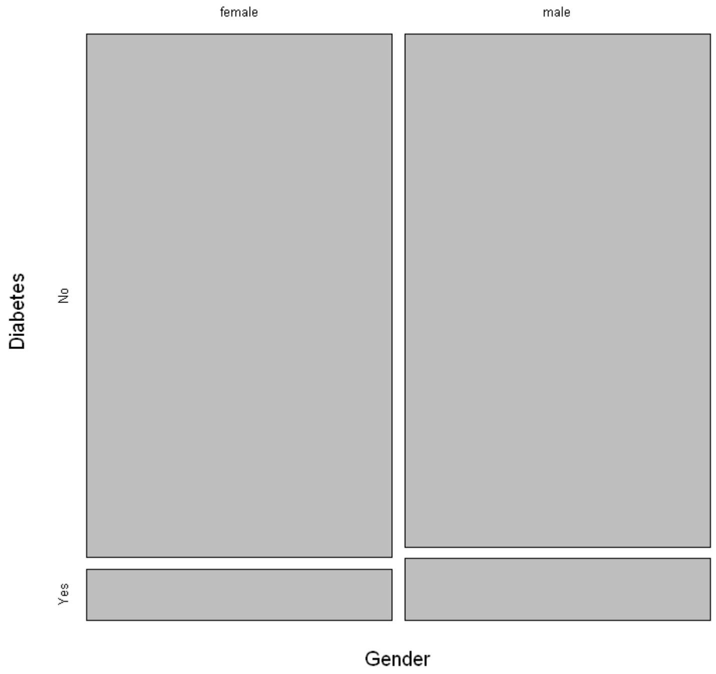
```
chisq.test(xt)
```

```
	Pearson's Chi-squared test with Yates' continuity correction

data:  xt
X-squared = 3.4332, df = 1, p-value = 0.0639
```

There’s a useful plot for visualizing contingency table data called a mosaic plot. Call the `mosaicplot()` function on the contingency table object.

In [101]:

```
mosaicplot(xt, main=NA)
```



10 Logistic regression

What if we wanted to model the discrete outcome, e.g., whether someone is insured, against several other variables, similar to how we did with multiple linear regression? We can't use linear regression because the outcome isn't continuous - it's binary, either Yes or No. For this we'll use logistic regression to model the $\log$ odds of binary response. That is, instead of modeling the outcome variable, Y, directly against the inputs, we'll model the log odds of the outcome variable.

If p is the probability that the individual is insured, then $\frac{p}{1-p}$ is the odds that person is insured. Then it follows that the linear model is expressed as:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k \tag{3}$$

Where β_0 is the intercept, β_1 is the increase in the odds of the outcome for every unit increase in x_1, and so on.

Logistic regression is a type of **generalized linear model** (GLM). We fit GLM models in R using the `glm ()` function. It works like the `lm ()` function except we specify which GLM to fit using the `family` argument. Logistic regression requires `family=binomial` . The typical use looks like this:

mod <- glm(y ~ x, data=yourdata, family='binomial') summary(mod)

Before we fit a logistic regression model let's relevel the Race variable so that "White" is the baseline. We saw above that people who identify as "White" have the highest rates of being insured. When we run the logistic regression, we'll get a separate coefficient (effect) for each level of the factor variable(s) in the model, telling you the increased odds that that level has, as compared to the baseline group.

In [102]:

```
nha$Race <- relevel(factor(nha$Race), ref="White")
xyz.Insured <- as.factor(nha$Insured)
head(xyz.Insured)
```

```
Yes  Yes  Yes  Yes  Yes  Yes
```

▶ **Levels**:

Now, let's fit a logistic regression model assessing how the odds of being insured change with different levels of race.

In [103]:

```
fit <- glm(xyz.Insured~Race, data=nha, family=binomial)
summary(fit)
```

```
Call:
glm(formula = xyz.Insured ~ Race, family = binomial, data = nha)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.0377   0.5177   0.5177   0.5177   1.1952  

Coefficients:
             Estimate Std. Error z value Pr(>|z|)    
(Intercept)   1.94218    0.06103  31.825  < 2e-16 ***
RaceAsian    -0.64092    0.17715  -3.618 0.000297 ***
RaceBlack    -0.59744    0.13558  -4.406 1.05e-05 ***
RaceHispanic -1.41354    0.14691  -9.622  < 2e-16 ***
RaceMexican  -1.98385    0.13274 -14.946  < 2e-16 ***
RaceOther    -1.26430    0.22229  -5.688 1.29e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3614.6  on 3704  degrees of freedom
Residual deviance: 3336.6  on 3699  degrees of freedom
  (2 observations deleted due to missingness)
AIC: 3348.6

Number of Fisher Scoring iterations: 4
```

The Estimate column shows the log of the odds ratio – how the log odds of having health insurance changes at each level of race compared to White. The P-value for each coefficient is on the far right. This shows that every other race has significantly less rates of health insurance coverage. But, as in our multiple linear regression analysis above, are there other important variables that we’re leaving out that could alter our conclusions? Lets add a few more variables into the model to see if something else can explain the apparent Race-Insured association. Let’s add a few things likely to be involved (Age and Income), and something that’s probably irrelevant (hours slept at night).

In [104]:

```
fit <- glm(xyz.Insured~Race+Age+Income+SleepHrsNight, data=nha, family=bin
summary(fit)
```

```
Call:
glm(formula = xyz.Insured ~ Race + Age + Income + SleepHrsNight,
    family = binomial, data = nha)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4815   0.3025   0.4370   0.6252   1.6871

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)   -3.501e-01  2.919e-01  -1.199    0.230
RaceAsian     -4.550e-01  2.031e-01  -2.241    0.025 *
RaceBlack     -2.387e-01  1.536e-01  -1.554    0.120
RaceHispanic  -1.010e+00  1.635e-01  -6.175 6.61e-10 ***
RaceMexican   -1.404e+00  1.483e-01  -9.468  < 2e-16 ***
RaceOther     -9.888e-01  2.422e-01  -4.082 4.46e-05 ***
Age            3.371e-02  2.949e-03  11.431  < 2e-16 ***
Income         1.534e-05  1.537e-06   9.982  < 2e-16 ***
SleepHrsNight -1.763e-02  3.517e-02  -0.501    0.616
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3284.3  on 3395  degrees of freedom
Residual deviance: 2815.0  on 3387  degrees of freedom
  (311 observations deleted due to missingness)
AIC: 2833

Number of Fisher Scoring iterations: 5
```

A few things become apparent:

1. Age and income are both highly associated with whether someone is insured. Both of these variables are highly significant $\left(P < 2.2 \times 10^{-16}\right)$, and the coefficient (the Estimate column) is positive, meaning that for each unit increase in one of these variables, the odds of being insured increases by the corresponding amount.
2. Hours slept per night is not meaningful at all.
3. After accounting for age and income, several of the race-specific differences are no longer statistically significant, but others remain so.
4. The absolute value of the test statistic (column called z value) can roughly be taken as an estimate of the "importance" of that variable to the overall model. So, age and income are the most important influences in this model; selfidentifying as Hispanic or Mexican are also very highly important, hours slept per night isn't important at all, and the other race categories fall somewhere in between.

Exercise 9 What’s the relationship between diabetes and participating in rigorous physical activity or sports?

```
-Create a contingency table with Diabetes status in rows and physical
 activity status in columns.
= Display that table with margins.
- Show the proportions of diabetics and nondiabetics, separately, who
 are physically active or not.
- Is this relationship significant?
- Create two different visualizations showing the relationship.
```

Model the same association in a logistic regression framework to assess the risk of diabetes using physical activity as a predictor.

```
- Fit a model with just physical activity as a predictor, and display a
 model summary.
- Add gender to the model, and show a summary.
- Continue adding weight and age to the model. What happens to the gende
r association?
- Continue and add income to the model. What happens to the original ass
ociation with physical activity?
```

11 Power & sample size

Statistical power, also sometimes called sensitivity, is defined as the probability that your test correctly rejects the null hypothesis when the alternative hypothesis is true. That is, if there really is an effect (difference in means, association between categorical variables, etc.), how likely are you to be able to detect that effect at a given statistical significance level, given certain assumptions. Generally there are a few moving pieces, and if you know all but one of them, you can calculate what that last one is.

1. Power: How likely are you to detect the effect? (Usually like to see 80% or greater).
2. N: What is the sample size you have (or require)?
3. Effect size: How big is the difference in means, odds ratio, etc?

If we know we want 80% power to detect a certain magnitude of difference between groups, we can calculate our required sample size. Or, if we know we can only collect 5 samples, we can calculate how likely we are to detect a particular effect. Or, we can work to solve the last one - if we want 80% power and we have 5 samples, what’s the smallest effect we can hope to detect?

All of these questions require certain assumptions about the data and the testing procedure. Which kind of test is being performed? What's the true effect size (often unknown, or estimated from preliminary data), what's the standard deviation of samples that will be collected (often unknown, or estimated from preliminary data), what's the level of statistical significance needed (traditionally $p < 0.05$, but must consider multiple testing corrections).

12 T-test power/N

The `power.t.test()` empirically estimates power or sample size of a t-test for differences in means. If we have 20 samples in each of two groups (e.g., control versus treatment), and the standard deviation for whatever we're measuring is 2.3, and we're expecting a true difference in means between the groups of 2, what's the power to detect this effect?

In [105]:

```
power.t.test(n=20, delta=2, sd=2.3)
```

```
     Two-sample t test power calculation 

              n = 20
          delta = 2
             sd = 2.3
      sig.level = 0.05
          power = 0.7641668
    alternative = two.sided

NOTE: n is number in *each* group
```

What's the sample size we'd need to detect a difference of 0.8 given a standard deviation of 1.5 , assuming we want 80% power?

In [106]:

```
power.t.test(power=.80, delta=.8, sd=1.5)
```

```
     Two-sample t test power calculation 

              n = 56.16413
          delta = 0.8
             sd = 1.5
      sig.level = 0.05
          power = 0.8
    alternative = two.sided

NOTE: n is number in *each* group
```

13 Proportions power/N

What about a two-sample proportion test (e.g., chi- square test)? If we have two groups (control and treatment), and we're measuring some outcome (e.g., infected yes/no), and we know that the proportion of infected controls is 80% but 20% in treated, what's the power to detect this effect in 5 samples per group?

In [107]:

```
power.prop.test(n=5, p1=0.8, p2=0.2)
```

```
     Two-sample comparison of proportions power calculation 

              n = 5
             p1 = 0.8
             p2 = 0.2
      sig.level = 0.05
          power = 0.4688159
    alternative = two.sided

NOTE: n is number in *each* group
```

How many samples would we need for 90% power?

In [108]:

```
power.prop.test(power=0.9, p1=0.8, p2=0.2)
```

```
     Two-sample comparison of proportions power calculation 

              n = 12.37701
             p1 = 0.8
             p2 = 0.2
      sig.level = 0.05
          power = 0.9
    alternative = two.sided

NOTE: n is number in *each* group
```

Also check out the pwr package which has power calculation functions for other statistical tests.

Function	Power calculations for
pwr.2p.test()	Two proportions (equal n)
pwr.2p2n.test()	Two proportions (unequal n)
pwr.anova.test()	Balanced one way ANOVA
pwr.chisq.test()	Chi-square test
pwr.f2.test()	General linear model
pwr.p.test()	Proportion (one sample)
pwr.r.test()	Correlation
pwr.t.test()	T-tests (one sample, 2 sample, paired)
pwr.t2n.test()	T-test (two samples with unequal n)

Exercise 10

1. You're doing a gene expression experiment. What's your power to detect a 2-fold change in a gene with a standard deviation of 0.7, given 3 samples? (Note - fold change is usually given on the $\log_2$ scale, so a 2 -fold change would be a delta of 1. That is, if the fold change is $2x$, then $\log_2(2) = 1$, and you should use 1 in the calculation, not 2).
2. How many samples would you need to have 80% power to detect this effect?
3. You're doing a population study looking at the effect of a SNP on disease X. Disease X has a baseline prevalence of 5% in the population, but you suspect the SNP might increase the risk of disease X by 10% (this is typical for SNP effects on common, complex diseases). How many samples do you need to have 80% power to detect this effect, given that you want a statistical significance of $p < 0.001$?