PM592: Regression Analysis for Health Data Science
Lab 2 – Data Manipulation & Statistics
Data Needed: *chs_dates.csv, chs_individual.csv, chs_regional.csv*

Outline

- Projects
- Merging & Setting Data
- Dates
- Statistical Tests: Z, T
- Factors

1. Projects

1.1. Purpose

Imagine you are working on several different projects at once. You have an analysis you have to run for a particular client, you are working on a set of interactive figures for a website, and you are working on analyses for PM592. By using Projects in RStudio, you can easily switch among workspaces and working directories.

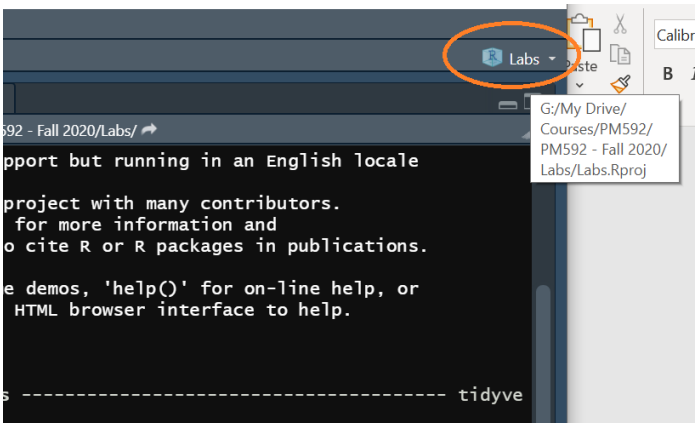
1.2. Creation

You can create an RStudio project in either a new directory, an existing directory that already has R code and data, or by cloning a version control (Git or Subversion) repository. For the purposes of this class, we will create a new project for PM592-related work.

- Determine the folder that has your PM592 code materials.
- Navigate to File > New Project
- Choose to create a project from an Existing Directory
- Choose the directory on your computer that has the required files.

1.3. Managing Projects

When you create a new project, there will be a .Rprofile, .RData, and .Rhistory file created in the specified directory. You can open the project directly or, within RStudio, use the Project menu at the top right to view your current project and switch between projects. The image below shows that I am in a project called "Labs" and shows me the directory associated with that project.



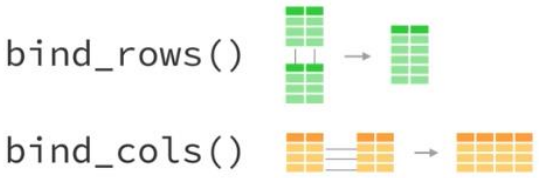
1.4. Opening a project will open all the .R files you had open when you last left the project. Unless

you had saved your workspace before exiting a project, your workspace will be empty when you switch to a new project (this means you'll have to re-load all data and packages). While saving your workspace may be convenient, this can take up a lot of space on your computer.

2. Merging & Setting Data

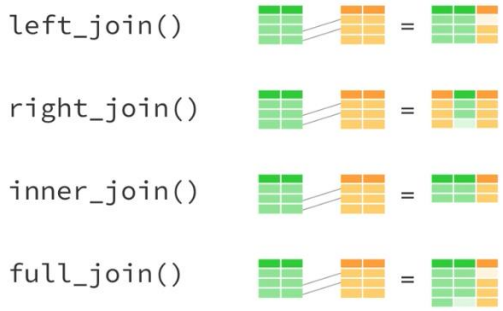
2.1. Dplyr provides extensive functionality with regard to combining two data sets.

2.2. A **bind** occurs when two datasets are assembled either by "taping together" their rows or columns.



- Example: you collected the same variables from different participants at two sites (UCLA and USC) and then need to connect these two data sets. You would `bind_rows`.

2.3. A **join** (aka merge) occurs when the information from two datasets is combined through some identifier variable.



2.3.1. A left_join will keep all observations that appear in the first data set.

2.3.2. A right_join will keep all observations that appear in the second data set.

2.3.3. An inner_join will keep only observations that appear in both data sets.

2.3.4. A full_join will keep all observations.

2.4. We have previously worked with the CHS data in lab. This file contains individual-level information, including performance in a pulmonary function test. Another file, named "chs_dates.csv," contains data on when the lung function test was performed.

2.5. In order to add the data collection date, we must merge the two data sets.

2.6. Both data sets have a variable "sid" in common—the identifier of each individual. This is the variable we will use to merge the two data sets. Let's verify that each file contains the same ID values. In the code below, we check both data sets to see if the number of IDs is the same and find that the "dates" file has 4 additional ID values.

```
length(chs_dates$sid)
[1] 1204

> length(chs_individual$sid)
[1] 1200
```

2.7. We now examine which values appear in one data set but not the other. The following code gives us the observation index of the individuals that appear in the chs_dates file, but not in the chs_individual file (there are 4 subjects). Then, it gives us the observation index of the individuals that appear in the chs_individual file, but not in the chs_dates file (there are 0 subjects).

```
> which(!(chs_dates$sid %in% chs_individual$sid))
[1]  10 403 500 923

> which(!(chs_individual$sid %in% chs_dates$sid))
integer(0)
```

2.8. Then let's find the ID values of individuals who appear in the "dates" file but not in the "individual" file. Persons 20, 691, 848, 1580 have date information but do not appear in the "individual" file.

```
> chs_dates$sid[which(!(chs_dates$sid %in% chs_individual$sid))]
[1]   20  691  848 1580
```

2.9. We are now at a crossroads. If we perform a full-join, we will include these 4 individuals in our new file, but there will not be any individual-level information for them.

```
> chs_individual %>%
+   full_join(chs_dates, by="sid") %>%
+   filter(sid %in% c(20, 691, 848, 1580))
# A tibble: 4 x 24
    sid townname  male race  hispanic agepft height weight   bmi asthma
  <dbl> <chr>    <dbl> <chr>    <dbl>  <dbl>  <dbl>  <dbl> <dbl>  <dbl>
1    20 NA          NA NA          NA     NA     NA     NA    NA     NA
2   691 NA          NA NA          NA     NA     NA     NA    NA     NA
3   848 NA          NA NA          NA     NA     NA     NA    NA     NA
4  1580 NA          NA NA          NA     NA     NA     NA    NA     NA
```

2.10. Instead, let's perform a left_join, keeping only participants who appear in the "individual" data set.

```
> chs_merged <-
+   chs_individual %>%
+   left_join(chs_dates, by="sid")
```

2.11. If you don't specify a join "by" variable, R will automatically choose the variable it thinks you want to merge by. It is good practice to explicitly specify which variable is your merge-by variable.

2.12. Refer to the following webpage for more information on binding and joining data. Figures in this section were taken from that site as well.
https://rpubs.com/williamsurles/293454

3. Dates

3.1. The Lubridate package provides increased functionality when working with dates in R.

3.2. Most statistical packages store dates as an integer value. In R, dates are stored as the number of

days that have elapsed since January 1, 1970.

3.3. In the following code, we transform the dates to their integer values and then display them to show how the dates are truly stored.

```
> chs_merged %>%
+   mutate(datenum = as.numeric(dt_collect)) %>%
+   select(datenum, dt_collect)
# A tibble: 1,200 x 2
   datenum dt_collect
     <dbl> <date>
 1   12785 2005-01-02
 2   12785 2005-01-02
 3   12787 2005-01-04
 4   12788 2005-01-05
 5   12788 2005-01-05
 6   12791 2005-01-08
 7   12791 2005-01-08
 8   12791 2005-01-08
 9   12793 2005-01-10
10   12793 2005-01-10
# ... with 1,190 more rows
```

3.4. Because we are using Lubridate, the "dt_collect" variable is stored as a special date type. There is a LOT of functionality available for date variables. One of the best aspects of Lubridate is that there is support for all of time's quirks such as time zones, leap days, and daylight savings time. You can learn more about them at https://lubridate.tidyverse.org/.

3.5. There are many functions that work with dates:

```
> chs_merged %>%
+   mutate(dow_collect = wday(dt_collect),
+          week_collect = week(dt_collect),
+          semester_collect = semester(dt_collect),
+          lyear_collect = leap_year(dt_collect)) %>%
+   select(ends_with("collect"))
# A tibble: 1,200 x 5
   dt_collect dow_collect week_collect semester_collect lyear_collect
   <date>           <dbl>        <dbl>            <int> <lgl>
 1 2005-01-02           1            1                1 FALSE
 2 2005-01-02           1            1                1 FALSE
 3 2005-01-04           3            1                1 FALSE
 4 2005-01-05           4            1                1 FALSE
 5 2005-01-05           4            1                1 FALSE
 6 2005-01-08           7            2                1 FALSE
 7 2005-01-08           7            2                1 FALSE
 8 2005-01-08           7            2                1 FALSE
 9 2005-01-10           2            2                1 FALSE
10 2005-01-10           2            2                1 FALSE
```

3.6. One of the most frequently encountered data manipulations for dates are intervals. Suppose we want to know how many months had passed between study start (1/1/2005) and data collection date. There are two ways that we can view this:

```
> chs_merged %>%
```

```
+   mutate(days_since_start1 = interval(ymd("2005-01-01"), dt_collect) / days(1),
+          days_since_start2 = ymd("2005-01-01") %--% dt_collect / days(1)) %>%
+   select(dt_collect, starts_with("days_"))
# A tibble: 1,200 x 3
   dt_collect days_since_start1 days_since_start2
   <date>                 <dbl>             <dbl>
 1 2005-01-02                 1                 1
 2 2005-01-02                 1                 1
 3 2005-01-04                 3                 3
 4 2005-01-05                 4                 4
 5 2005-01-05                 4                 4
 6 2005-01-08                 7                 7
 7 2005-01-08                 7                 7
 8 2005-01-08                 7                 7
 9 2005-01-10                 9                 9
10 2005-01-10                 9                 9
```

4. Statistical Tests: Z & t

4.1. One Sample Z-test

4.1.1. FEV1 is the volume of air that can be expired through a single breath in one second. Suppose it is known that the mean FEV for adolescents in the United States is 2,055mL, with a SD=330mL. Is the mean FEV of adolescents in Southern California different from the national average?

4.1.2. R does not have a built-in z-test. We would have to calculate this by hand:

```
> mean(chs_merged$fev, na.rm=T)
[1] 2031.265

> zstat <- (mean(chs_merged$fev, na.rm=T) - 2055)/
+   (330/sqrt(length(!is.na(chs_merged$fev))))

> pnorm(1-2*abs(zstat))
[1] 3.402825e-05
```

4.1.3. The mean FEV of adolescents in Southern California is 2,031mL, which is significantly lower than 2,055 (the mean FEV of adolescents in the United States) (p<.001).

4.2. One Sample t-test

4.2.1. Consider the question in the previous section. Let's perform a t-test of the same research question, assuming we don't know the population standard deviation.

```
> t.test(chs_merged$fev, mu=2055)

      One Sample t-test

data:  chs_merged$fev
t = -2.386, df = 1104, p-value = 0.0172
alternative hypothesis: true mean is not equal to 2055
95 percent confidence interval:
 2011.747 2050.783
sample estimates:
mean of x
 2031.265
```

4.2.2. The mean FEV of adolescents in Southern California is 2,031mL, which is significantly lower than the mean FEV of adolescents in the United States (p=.017).

4.3. Two Sample t-test

4.3.1. Does the mean FEV of adolescents in Southern California vary by gender?

4.3.2. The type of t-test performed depends on equality of variance between two groups. Let's perform a test for equality of variances.

```
> var.test(fev ~ male, data=chs_merged)

      F test to compare two variances

data:  fev by male
F = 1.058, num df = 553, denom df = 550, p-value = 0.5085
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.8951431 1.2503363
sample estimates:
ratio of variances
          1.057957
```

4.3.3. The variance of FEV does not differ between males and females (p=.51), so we can proceed with the assumption of equal variance.

```
> t.test(fev ~ male, var.equal = T, data=chs_merged)

      Two Sample t-test

data:  fev by male
t = -7.4514, df = 1103, p-value = 1.861e-13
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -182.8212 -106.6080
sample estimates:
mean in group 0 mean in group 1
       1959.105        2103.819
```

4.3.4. The mean FEV of males in Southern California is 2,104mL, while the mean for females is 1,959mL. A t-test showed that these means are statistically significantly different (t_{1103}=-7.45, p<.001).

4.4. Paired Samples t-test

4.4.1. If the observations between groups are linked (i.e., non-independent) then we can use a paired-samples t-test. Use the "paired = TRUE" option in the t.test function.

4.4.2. Example: suppose for each individual we measured FEV in the morning and then again in the evening. To measure the change in FEV from morning to evening, we would use a paired samples t-test. This is because the individuals who are measured in the morning are the same individuals who are measured in the evening; the two samples are not independent.

4.5. The htest class object

4.5.1. The t.test, var.test, and many base R test functions return an object of class htest. It is a

list object that contains information about the hypothesis test being performed. You can extract information such as test statistics and p-values from this object.

```
> ttest_fevmale <-
+   t.test(fev ~ male, var.equal = F, data=chs_merged)

> class(ttest_fevmale)
[1] "htest"

> str(ttest_fevmale)
List of 10
 $ statistic  : Named num -7.45
  ..- attr(*, "names")= chr "t"
 $ parameter  : Named num 1102
  ..- attr(*, "names")= chr "df"
 $ p.value    : num 1.85e-13
 $ conf.int   : num [1:2] -183 -107
  ..- attr(*, "conf.level")= num 0.95
 $ estimate   : Named num [1:2] 1959 2104
  ..- attr(*, "names")= chr [1:2] "mean in group 0" "mean in group 1"
 $ null.value : Named num 0
  ..- attr(*, "names")= chr "difference in means"
 $ stderr     : num 19.4
 $ alternative: chr "two.sided"
 $ method     : chr "Welch Two Sample t-test"
 $ data.name  : chr "fev by male"
 - attr(*, "class")= chr "htest"

> ttest_fevmale$statistic
       t
-7.451955

> ttest_fevmale$p.value
[1] 1.853947e-13
```

4.6. Assessing Normality

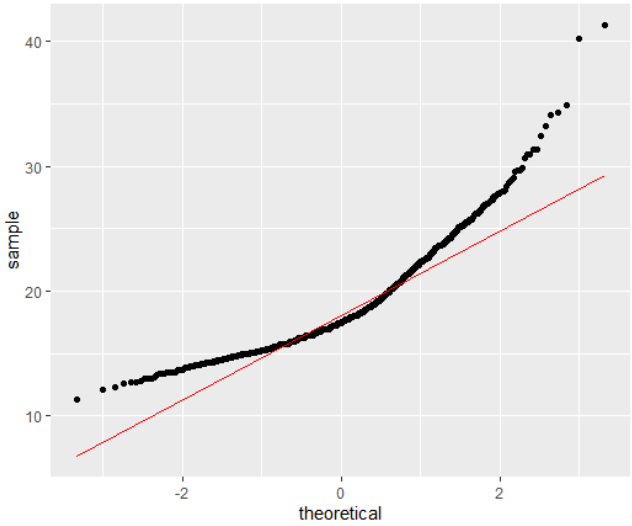
4.6.1. Statistical tests such as the Shapiro-Wilk and Kolmogorov-Smirnov tests can be used to detect departures from normality. The null hypothesis is that the data is distributed normally, while the alternative hypothesis is that the distribution departs from normality. We see that the variable BMI is not normally distributed (p<.001).

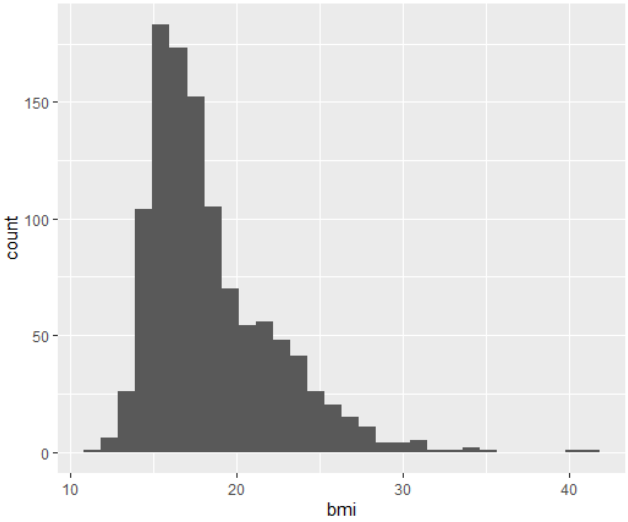
```
> shapiro.test(chs_merged$bmi)

      Shapiro-Wilk normality test

data:  chs_merged$bmi
W = 0.89333, p-value < 2.2e-16
```

4.6.2. We can use visual confirmation from histograms and QQ plots.

```
> chs_merged %>%
+   ggplot(aes(sample=bmi)) +
+   geom_qq() +
```

```
+   geom_qq_line(color = "red")

>
> chs_merged %>%
+   ggplot(aes(x=bmi)) +
+   geom_histogram()
```



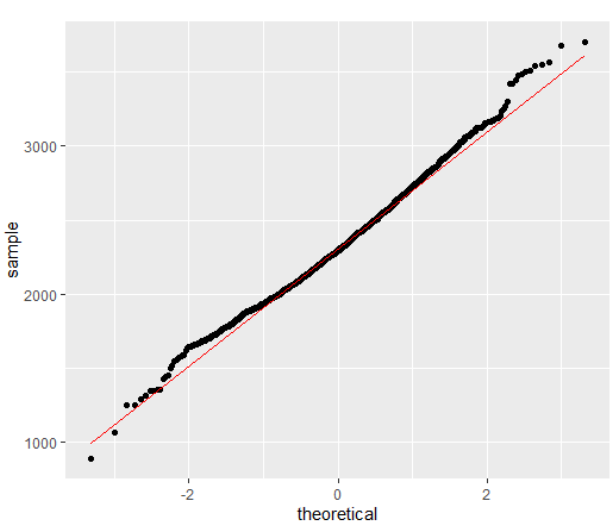


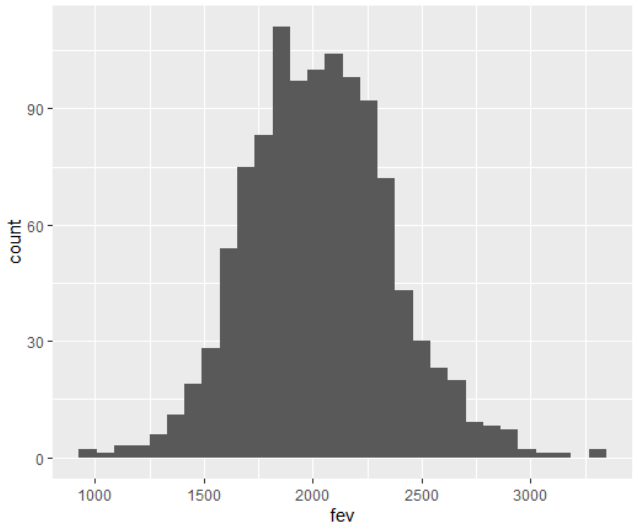
4.6.3. The distribution of BMI does appear skewed.

4.6.4. Recall that if the sample size is large enough (n > 30 or so) then we can assume the sampling distribution of the mean is normal, and proceed with a parametric test.

4.6.5. When the sample size is quite large, tests of normality are sensitive to departures from normality. If we look at the distribution of FEV it appears relatively normal, but the Shapiro-Wilk test still rejects the null hypothesis of normality (p=.001). Luckily in these situations the central limit theorem lets us proceed with statistical testing anyway.

```
> chs_merged %>%
+   ggplot(aes(sample=fvc)) +
+   geom_qq() +
+   geom_qq_line(color = "red")

> chs_merged %>%
+   ggplot(aes(x=fev)) +
+   geom_histogram()

> shapiro.test(chs_merged$fev)

      Shapiro-Wilk normality test

data:  chs_merged$fev
W = 0.99501, p-value = 0.00105
```





4.6.6. Yet, if we sample 20 individuals from this data set and run the Shapiro-Wilk test, we do not find a departure from normality. This reinforces the sensitivity of normality tests at large sample sizes.

```
> shapiro.test(sample(chs_merged$fev, 20))

      Shapiro-Wilk normality test

data:  sample(chs_merged$fev, 20)
W = 0.94579, p-value = 0.3343
```

- Use the "read_csv" function to read-in the chs_individual.csv file.

4.6.7. Read more about different types of normality tests here: (https://www.tandfonline.com/doi/pdf/10.1080/00949655.2010.520163)

4.7. Non-parametric Tests

4.7.1. The Z- and t-tests rely on the assumption that the sampling distribution of the mean follows a t- or Z- distribution.

4.7.2. When the sample size is small and your outcome is non-normally distributed, it will be necessary to run a non-parametric test.

4.7.3. Non-parametric tests do not violate any distributional assumptions, but they will lead to lower power if the data actually is normally distributed.

4.7.4. The Mann-Whitney test (aka Wilcoxon rank sum test) is used for independent groups.

```
> wilcox.test(fev ~ male, data=chs_merged)

      Wilcoxon rank sum test with continuity correction

data:  fev by male
W = 113678, p-value = 2.089e-13
alternative hypothesis: true location shift is not equal to 0
```

4.7.5. The Wilcoxon Sign-Rank test is used when the groups are paired (analogous to a paired t-test). Use the "paired=TRUE" option.

5. Factors

5.1. Purpose of factors

5.1.1. Factor variables are categorical and behave differently in some functions compared to character or numeric variables. They will be treated as unordered (or ordered, if specified) categories as opposed to continuous values.

5.2. Factor labels

5.2.1. Factorizing a variable also provides label information for its values. Suppose you have a variable called "sex" that takes on values of 1 or 2; it is not readily apparent which value corresponds to male and which value corresponds to female.

5.2.2. We can specify the levels of a factor and the values these levels taken on. Here, we create a new "male.f" variable (male factor) that provides information about the sex of the participant. Then we make sure that all values in the "male.f" variable are correctly assigned based on the "male" variable.

```
> chs_merged <-
+   chs_merged %>%
+   mutate(male.f = factor(male,
+                          levels = c(0, 1),
+                          labels = c("Female", "Male")))

> chs_merged %>%
+   count(male.f, male)
# A tibble: 2 x 3
  male.f  male     n
  <fct>  <dbl> <int>
1 Female     0   610
2 Male       1   590
```

5.2.3. This may help with interpreting certain output. Let's see how it would affect our t-test that we previously ran.

```
> t.test(fev ~ male.f, var.equal = F, data=chs_merged)

      Welch Two Sample t-test

data:  fev by male.f
t = -7.452, df = 1102.4, p-value = 1.854e-13
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -182.8183 -106.6109
sample estimates:
mean in group Female   mean in group Male
            1959.105             2103.819
```

Lab 2 Exercises

Objective(s):	Merge two data sets, create factor variables, assess variable normality, practice variable manipulation
Datasets Required:	chs_individual, chs_regional

1. The CHS_INDIVIDUAL data contains information on participants' demographics and lung function. However, another data set (CHS_REGIONAL) contains information on air pollution in the towns that these participants come from. Read in both data sets.

    a. Are there values of townname in the INDIVIDUAL data that do not appear in the REGIONAL data?

        i. no

    b. Are there values of townname in the REGIONAL data that do not appear in the INDIVIDUAL data?

        i. no

    c. Merge merge both data sets by "townname" and verify that the merge was successful.

        i. The merged data set has 1200 observations, as expected

```
chs_merged <- chs_individual %>% left_join(chs_regional,
by="townname")
```

2. Using the CHS codebook, create factor variables for parental education and Hispanic ethnicity.

```
chs_merged <- chs_merged %>%
+   mutate(hispanic.f = factor(hispanic, levels=c(0,1), labels=c("Non-Hispanic", "Hispanic
"))) %>%
+   mutate(educ_parent.f = factor(educ_parent,
+                                 levels=c(1, 2, 3, 4, 5),
+                                 labels=c("<12 Grade",
+                                          "Grade 12",
+                                          "Some post high school",
+                                          "4 years of college",
+                                          "Some post-graduate")))
```

    a. Verify that the values of parental education are correct.

```
  educ_parent educ_parent.f             n
        <dbl> <fct>                 <int>
1           1 <12 Grade               169
2           2 Grade 12                227
3           3 Some post high school   516
4           4 4 years of college      114
5           5 Some post-graduate      110
6          NA NA                       64
> # b
```

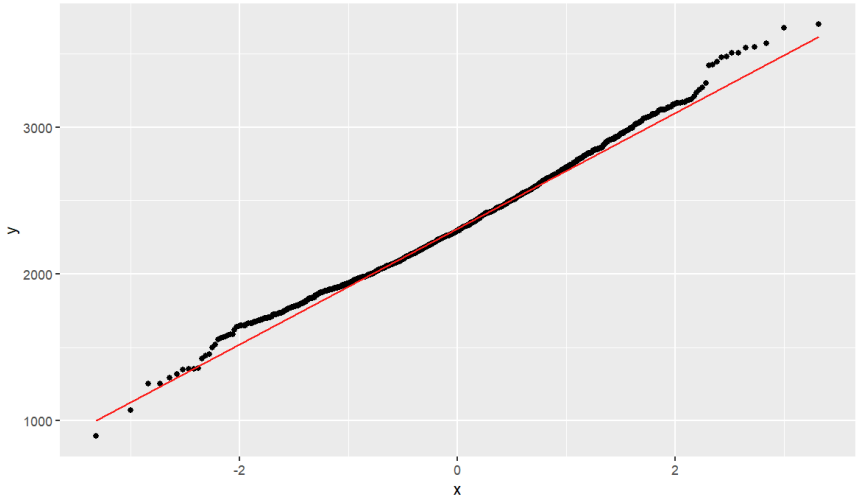
    b. Verify that the values of Hispanic ethnicity are correct.

```
# A tibble: 2 × 3
  hispanic hispanic.f       n
     <dbl> <fct>        <int>
1        0 Non-Hispanic   679
```

```
2        1 Hispanic       521
```

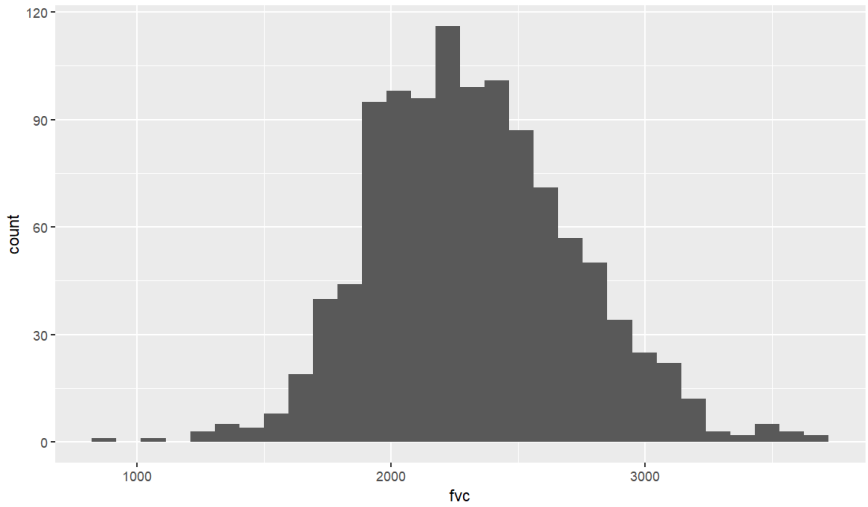
3. Assess the normality of the FVC variable. FVC ("forced vital capacity") is the total amount of air that can be exhaled after taking the deepest breath possible.

   a. Does FVC appear normal based on a QQ plot?



   The QQ plot does seem to fit a straight line, the edges depart from it a bit which happens in practice

   b. Does FVC appear normal based on a histogram?



   The histogram appears to be about normal

   c. Does FVC appear normal based on a normality test?

```
> shapiro.test(chs_merged$fvc) # p < 0.001 rejects null hypothesis of normality

     Shapiro-Wilk normality test

data:  chs_merged$fvc
W = 0.99256, p-value = 2.406e-05
```

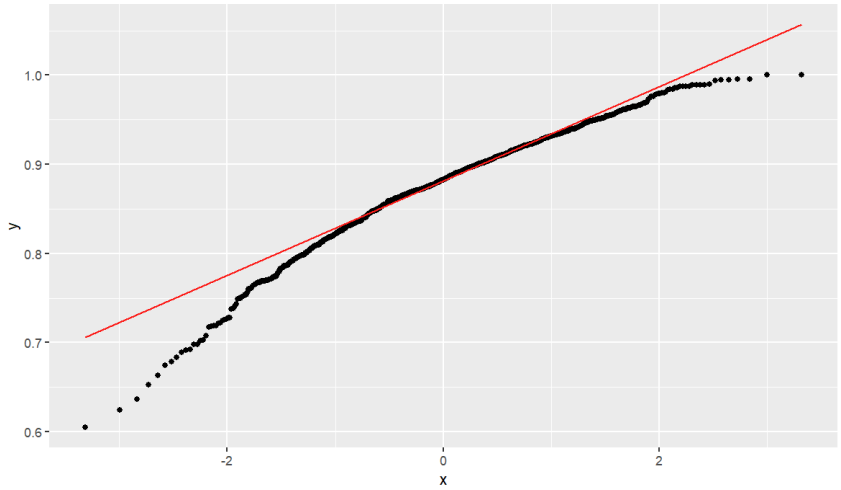
4. The ratio of FEV to FVC is another indicator of lung health, with higher values being more desirable. Create

a new variable that is equal to FEV/FVC. Then, fill out the number (and percent of the sample) that fall into each of the following categories of lung function according to the following table:

RANGE	CATEGORY	N (%)
>= 70%	Normal	1087 (90.6%)
60-69%	Mild Deficiency	13 (1.08%)
50-59%	Moderate Deficiency	0 (0%)
<50%	Severe Deficiency	0 (0%)

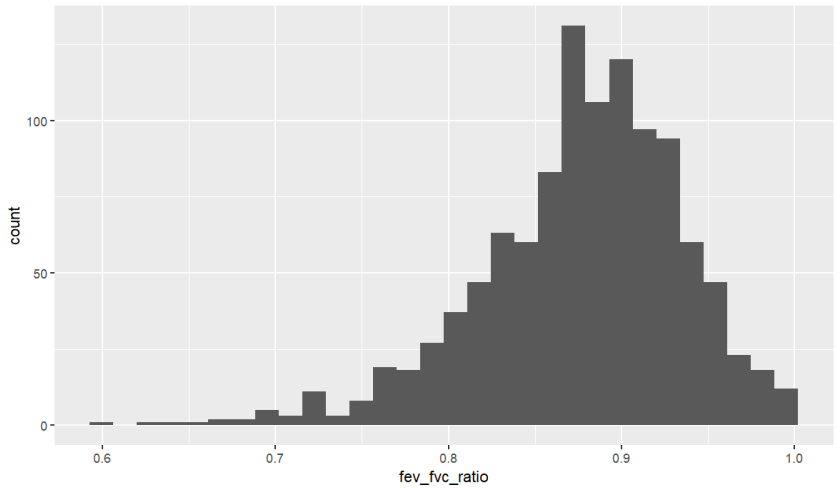
5. Assess the normality of the FEV/FVC ratio variable.

   a. Does FEV/FVC appear normal based on a QQ plot?



The FEV/FVC ratio variable does not appear normal on a QQ plot

   b. Does FEV/FVC appear normal based on a histogram?



It appears to be skewed left

   c. Does FEV/FVC appear normal based on a normality test?

```
> shapiro.test(chs_merged$fev_fvc_ratio) # p << 0.001 rejects null hypothesis of normality

    Shapiro-Wilk normality test
```

```
data:  chs_merged$fev_fvc_ratio
W = 0.96418, p-value = 8.185e-16
```

6. Does the ratio of FEV/FVC significantly vary based on (provide your output, the test statistic, and p-value):

d. Sex (male vs. female)?

```
> t.test(fev_fvc_ratio ~ as.factor(male), var.equal = T, data=chs_merged)

     Two Sample t-test

data:  fev_fvc_ratio by as.factor(male)
t = 5.8394, df = 1098, p-value = 6.894e-09
alternative hypothesis: true difference in means between group 0 and group 1 is not equal
to 0
95 percent confidence interval:
 0.01354005 0.02724418
sample estimates:
mean in group 0 mean in group 1
      0.8867891       0.8663970
```

With a p-value less than 0.01, we have sufficient evidence to reject the null hypothesis that the mean of FEV/FVC is the same in males and females, and accept the alternative hypothesis that the mean of FEV/FVC is different in males than in females.

e. Ethnicity (Hispanic vs. non-Hispanic)?

```
> t.test(fev_fvc_ratio ~ hispanic.f, var.equal = T, data=chs_merged)

     Two Sample t-test

data:  fev_fvc_ratio by hispanic.f
t = -0.74831, df = 1098, p-value = 0.4544
alternative hypothesis: true difference in means between group Non-Hispanic and group Hisp
anic is not equal to 0
95 percent confidence interval:
 -0.009678626  0.004334384
sample estimates:
mean in group Non-Hispanic     mean in group Hispanic
                 0.8754173                  0.8780894
```

With a p-value greater than 0.01, we do not have sufficient evidence to reject the null hypothesis that the mean of FEV/FVC is the same in Hispanics vs non-Hispanics.

f. Asthma status (asthmatic vs. no-asthma)?

```
> t.test(fev_fvc_ratio ~ as.factor(asthma), var.equal = F, data=chs_merged)

     Welch Two Sample t-test

data:  fev_fvc_ratio by as.factor(asthma)
t = 6.6294, df = 200.18, p-value = 3.063e-10
alternative hypothesis: true difference in means between group 0 and group 1 is not equal
to 0
95 percent confidence interval:
 0.02565569 0.04737980
sample estimates:
mean in group 0 mean in group 1
      0.8819433       0.8454256
```

With a p-value less than 0.01, we have sufficient evidence to reject the null hypothesis that the mean of FEV/FVC is the same in children with asthma and children without asthma, and accept the

alternative hypothesis that the mean of FEV/FVC is different in children with asthma and children without asthma.