431 Class 12

Thomas E. Love

2017-10-05

Today's Agenda

1. Some Thoughts on `dplyr` and its verbs
2. The Printer Case Study
3. Dealing with Missing Data via Imputation
4. Setting up the first Quiz

Setting Up Quiz 1

There are a total of 41 questions, 18 worth 2 points, 18 worth 3 points, 4 worth 2.5 points, and 1 that affirms your work is yours alone.

- Please select or type in your best response for each question. The questions are not arranged in any particular order, and you should answer all of them.
- You must complete this quiz by Noon on Monday, 2016-10-09. You will have the opportunity to edit your responses after completing the quiz, but this must be completed by the deadline.
- If you wish to complete part of the quiz and then return to it later, please scroll down to the end of the quiz and complete the **affirmation** (Question 41). The affirmation is required, and you will have to complete it in order to exit the quiz and save your progress. You will then be presented with a link to "Edit your progress" which you will want to bookmark, so you can return to it easily.

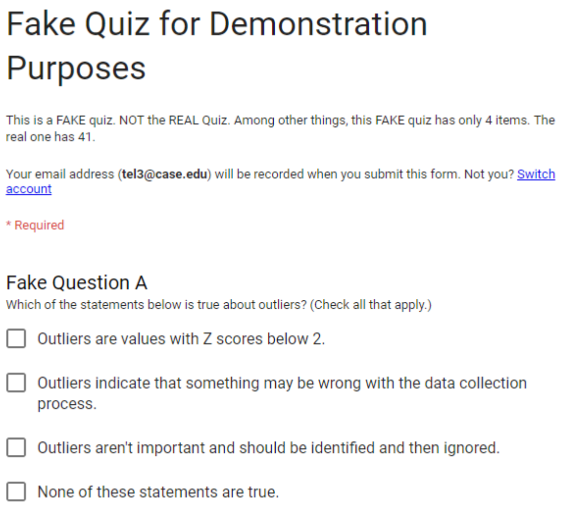
Quiz 1: Main item types.

Fake Quiz is at https://goo.gl/forms/hw37w3BrpibPDGQ03

1. Short Answer Questions
2. Multiple Choice
3. Checkboxes
4. Matching

- You are welcome to consult the materials provided on the course website, but you are not allowed to discuss the questions on this quiz with anyone other than Professor Love or the Teaching Assistants, who may be reached at `431-help at case dot edu`.

Fake Quiz: Question A



Fake Quiz for Demonstration Purposes

This is a FAKE quiz. NOT the REAL Quiz. Among other things, this FAKE quiz has only 4 items. The real one has 41.

Your email address (**tel3@case.edu**) will be recorded when you submit this form. Not you? Switch account

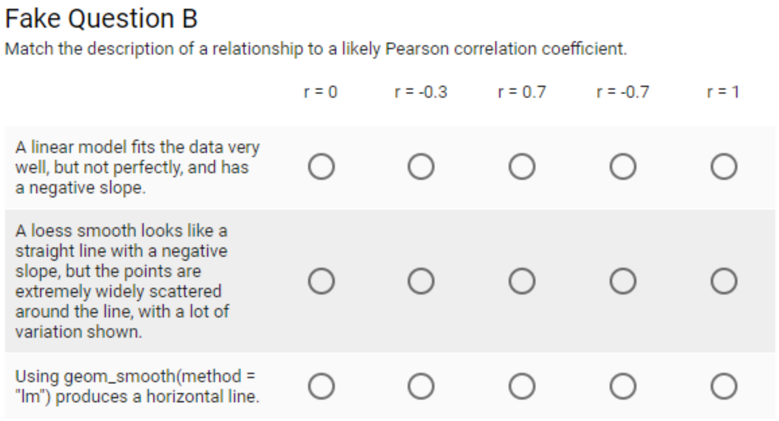
* Required

Fake Question A

Which of the statements below is true about outliers? (Check all that apply.)

- ☐ Outliers are values with Z scores below 2.
- ☐ Outliers indicate that something may be wrong with the data collection process.
- ☐ Outliers aren't important and should be identified and then ignored.
- ☐ None of these statements are true.

Fake Quiz: Question B

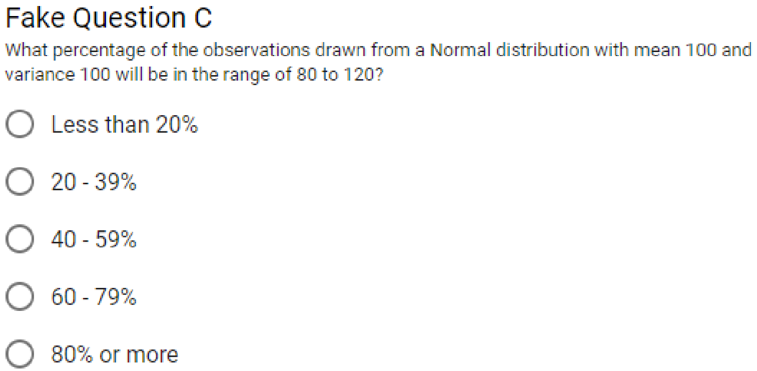


Fake Question B

Match the description of a relationship to a likely Pearson correlation coefficient.

	r = 0	r = -0.3	r = 0.7	r = -0.7	r = 1
A linear model fits the data very well, but not perfectly, and has a negative slope.	○	○	○	○	○
A loess smooth looks like a straight line with a negative slope, but the points are extremely widely scattered around the line, with a lot of variation shown.	○	○	○	○	○
Using geom_smooth(method = "lm") produces a horizontal line.	○	○	○	○	○

Fake Quiz: Question C

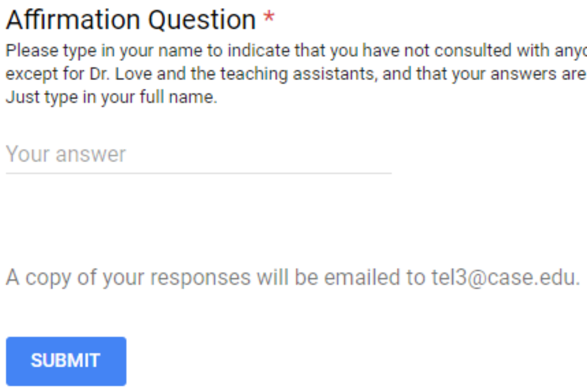


Fake Question C

What percentage of the observations drawn from a Normal distribution with mean 100 and variance 100 will be in the range of 80 to 120?

- ○ Less than 20%
- ○ 20 - 39%
- ○ 40 - 59%
- ○ 60 - 79%
- ○ 80% or more

Fake Quiz: Affirmation

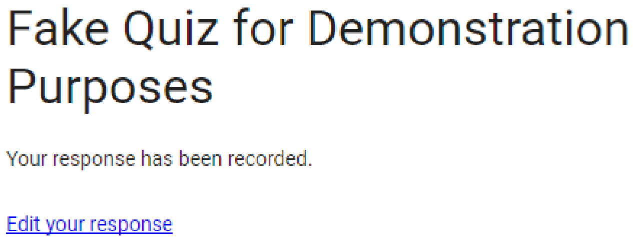

Affirmation Question *

Please type in your name to indicate that you have not consulted with anyone else about this quiz except for Dr. Love and the teaching assistants, and that your answers are yours and yours alone. Just type in your full name.

Your answer

A copy of your responses will be emailed to tel3@case.edu.

Fake Quiz: Completion


Fake Quiz for Demonstration Purposes

Your response has been recorded.

Edit your response

Today's R Setup

```r
library(mice); library(tidyverse)

source("Love-boost.R")
```

`dplyr` basics: The Key Verbs

Six key functions:

- Pick observations by their values (`filter()`).
- Reorder the rows (`arrange()`).
- Pick variables by their names (`select()`).
- Collapse many values down to a single summary (`summarise()`).
- Create new variables with functions of existing variables (`mutate()`).
- Change the scope of another function from operating on the whole data set to operating on it group-by-group (`group_by()`)

All of this comes from Wickham and Grolemund, R for Data Science, Chapter 5

http://r4ds.had.co.nz/transform.html#introduction-2

dplyr basics: How the verbs work

- The first argument is a data frame (or tibble).
- The second arguments describe what to do with the data frame. You can refer to columns in the data frame directly without using $.
- The result is a new data frame.

We'll work with the wcgs data.

```
wcgs <- read.csv("wcgs.csv") %>% tbl_df
wcgs
```

```
# A tibble: 3,154 x 22
      id   age   agec height weight   lnwght wghtcat
   <int> <int> <fctr>  <int>  <int>    <dbl>  <fctr>
1   2343    50  46-50     67    200 5.298317 170-200
2   3656    51  51-55     73    192 5.257495 170-200
3   3526    59  56-60     70    200 5.298317 170-200
4  22057    51  51-55     69    150 5.010635 140-170
```

Filter rows with `filter()`

`filter()` allows you to subset observations based on their values.

```
wcgs.sub1 <- wcgs %>%
  filter(dibpat == "Type A" & age > 49)
wcgs.sub1
```

```
# A tibble: 522 x 22
      id   age   agec height weight   lnwght wghtcat
   <int> <int> <fctr>  <int>  <int>    <dbl>  <fctr>
 1  2343    50  46-50     67    200 5.298317 170-200
 2  3656    51  51-55     73    192 5.257495 170-200
 3  3526    59  56-60     70    200 5.298317 170-200
 4 22057    51  51-55     69    150 5.010635 140-170
 5 12681    50  46-50     71    195 5.273000 170-200
 6  3284    59  56-60     72    206 5.327876   > 200
 7 21071    54  51-55     67    152 5.023880 140-170
 8 13371    55  51-55     72    185 5.220356 170-200
```

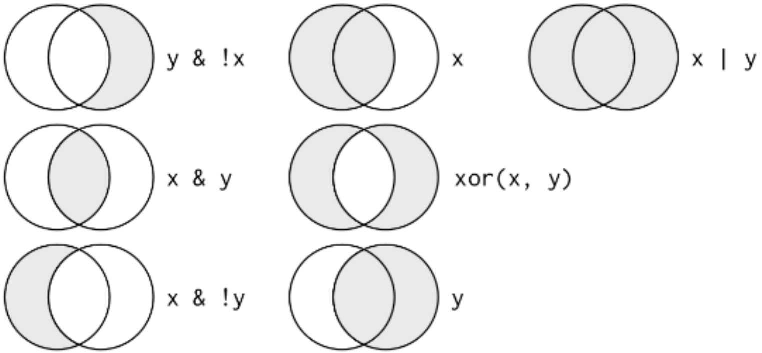
Comparison and Logical Operators

Comparison Operator	Meaning
>	is greater than
>=	is greater than or equal to
<	is less than
<=	is less than or equal to
!=	is not equal to
==	is equal to

Logical (Boolean) Operator	Meaning
&	and
|	or
!	not

Missing Values (`NA` in R) can make things tricky. They are contagious. Almost any operation involving an unknown value will also be unknown.

The complete set of Boolean Operators



Source: http://r4ds.had.co.nz/transform.html#logical-operators

Arrange rows with `arrange()`

`arrange()`, instead of selecting rows (like `filter()`), changes their order.

- Use `arrange(height)` to arrange in ascending order of height. Provide a second column name to break ties, if you like.
- Missing values are always sorted at the end.

```
wcgs %>%
  arrange(desc(height), desc(weight))
```

```
# A tibble: 3,154 x 22
     id   age   agec height weight   lnwght wghtcat
  <int> <int> <fctr>  <int>  <int>    <dbl>  <fctr>
1 12012    47  46-50     78    250 5.521461   > 200
2  2145    41  41-45     78    220 5.393628   > 200
3 12680    43  41-45     78    190 5.247024 170-200
4 13512    42  41-45     77    220 5.393628   > 200
5 12620    49  46-50     77    210 5.347107   > 200
6 11209    45  41-45     77    195 5.273000 170-200
```

Select columns with `select()`

`select()` lets you zoom in on the columns you actually want to use based on the names of the variables. R for Data Science lays out some helper functions within select() for use in bigger data sets.

```
wcgs.sub2 <- wcgs %>%
  select(id, age, height, weight, dibpat, smoke, behpat)
wcgs.sub2
```

```
# A tibble: 3,154 x 7
     id   age height weight dibpat  smoke behpat
  <int> <int>  <int>  <int> <fctr> <fctr> <fctr>
1  2343    50     67    200 Type A    Yes     A1
2  3656    51     73    192 Type A    Yes     A1
3  3526    59     70    200 Type A     No     A1
4 22057    51     69    150 Type A     No     A1
5 12927    44     71    160 Type A     No     A1
6 16029    47     64    158 Type A    Yes     A1
```

Grouped summaries with `summarize()`

`summarise()` or `summarize()` collapses a data frame to a single row.

```
wcgs.sub2 %>%
  summarize(mean.ht = mean(height, na.rm=TRUE),
            sd.ht = sd(height, na.rm=TRUE)) %>%
  round(digits = 2)
```

```
# A tibble: 1 x 2
  mean.ht sd.ht
    <dbl> <dbl>
1   69.78  2.53
```

Using the pipe (%>%) to filter and summarize

```
wcgs.sub2 %>%
 filter(dibpat == "Type A") %>%
 summarize(pearson.r = cor(height, weight),
  spearman.r = cor(height, weight, method = "spearman")) %>%
 round(digits = 3) %>%
 knitr::kable()
```

pearson.r	spearman.r
0.534	0.542

Using group_by() with summarize to look group-by-group

```r
wcgs.sub2 %>%
  group_by(behpat) %>%
  summarize(
    pearson.r = round(cor(height, weight),3) ) %>%
  knitr::kable()
```

behpat	pearson.r
A1	0.571
A2	0.526
B3	0.524
B4	0.557

Using group_by() to look at separated groups

You might have tried this approach instead, but it throws an error...

```
wcgs.sub2 %>%
  group_by(behpat) %>%
  summarize(
    pearson.r = cor(height, weight)) %>%
  round(digits = 3) %>%
  knitr::kable()
```

- Why doesn't this work?

Using group_by() to look at separated groups

You might have tried this approach instead, but it throws an error...

```
wcgs.sub2 %>%
  group_by(behpat) %>%
  summarize(
    pearson.r = cor(height, weight)) %>%
  round(digits = 3) %>%
  knitr::kable()
```

- Why doesn't this work?
- When R sees the round command, it tries to apply it to every element of the table, including the behavior pattern labels, which aren't numbers. So it throws an error.

Add new variables with `mutate()`

`mutate()` adds new columns that are functions of existing columns to the end of your data set.

Suppose we want to calculate the weight/height ratio for each subject.

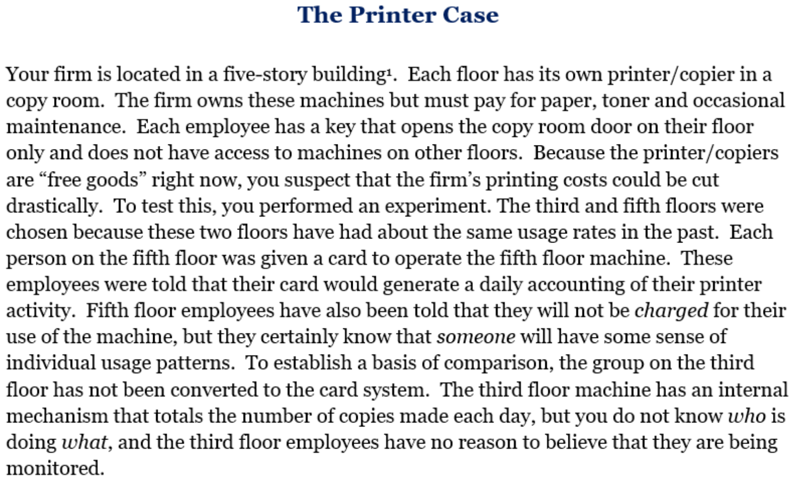
```
wcgs.sub3 <- wcgs.sub2 %>%
  mutate(wh.ratio = weight / height)
wcgs.sub3
```

```
# A tibble: 3,154 x 8
      id   age height weight dibpat  smoke behpat
   <int> <int>  <int>  <int> <fctr> <fctr> <fctr>
 1  2343    50     67    200 Type A    Yes     A1
 2  3656    51     73    192 Type A    Yes     A1
 3  3526    59     70    200 Type A     No     A1
 4 22057    51     69    150 Type A     No     A1
 5 12927    44     71    160 Type A     No     A1
 6 16029    47     64    158 Type A    Yes     A1
```

On Coding and dplyr

1. Learn `dplyr`, and use it to do most of your data management within R.
    - `dplyr` is mostly about these key verbs, and piping, for our purposes
    - some tasks produce results which be confusing, we're here to help
2. `dplyr` is most useful in combination with other elements of the `tidyverse`, most prominently `ggplot2`.
3. `Hmisc` doesn't play nicely with `dplyr`, so don't load the whole Hmisc library, just call individual functions you need with, for example, `Hmisc::describe` or `Hmisc::smean.cl.boot`

The Printer Case, Setup



The Printer Case

Your firm is located in a five-story building[1]. Each floor has its own printer/copier in a copy room. The firm owns these machines but must pay for paper, toner and occasional maintenance. Each employee has a key that opens the copy room door on their floor only and does not have access to machines on other floors. Because the printer/copiers are "free goods" right now, you suspect that the firm's printing costs could be cut drastically. To test this, you performed an experiment. The third and fifth floors were chosen because these two floors have had about the same usage rates in the past. Each person on the fifth floor was given a card to operate the fifth floor machine. These employees were told that their card would generate a daily accounting of their printer activity. Fifth floor employees have also been told that they will not be *charged* for their use of the machine, but they certainly know that *someone* will have some sense of individual usage patterns. To establish a basis of comparison, the group on the third floor has not been converted to the card system. The third floor machine has an internal mechanism that totals the number of copies made each day, but you do not know *who* is doing *what*, and the third floor employees have no reason to believe that they are being monitored.

The Printer Case, Main Table

You collected data from the machines over the last 50 working days. The data are in the table below and can be downloaded from the web in the **printer.csv** file. There are three variables: DAY, which indicates the day; FIFTH, the number of copies made on the 5th floor; and THIRD, the number made on the 3rd floor.

Will the card accounting system effectively lower usage if implemented across the firm?

Day	Fifth	Third	Day	Fifth	Third	Day	Fifth	Third	Day	Fifth	Third
1	750	340	14	570	370	27	390	270	39	270	400
2	710	540	15	570	720	28	420	670	40	250	130
3	700	210	16	560	670	29	380	660	41	210	440
4	720	530	17	500	460	30	370	240	42	240	130
5	690	550	18	480	320	31	370	500	43	190	250
6	670	350	19	550	370	32	360	480	44	160	330
7	660	590	20	510	570	33	350	560	45	130	300
8	640	520	21	520	120	34	330	310	46	120	110
9	670	360	22	460	190	35	280	390	47	180	740
10	620	420	23	470	710	36	300	610	48	150	700
11	580	160	24	440	620	37	310	690	49	110	150
12	590	470	25	400	180	38	290	410	50	100	580
13	610	380	26	410	640						

The Printer Case Discussion, Part 1

Fifty days of data:

- Fifth floor employees were given a card to operate their printer.
- Third floor employees were not.

1. Is this a randomized trial or an observational study?
2. What is the outcome we are studying?
3. What are the two treatments/exposures/interventions being compared?
4. What controls are in place as part of the study's design?
5. **Key Question**: Will the card accounting system effectively lower usage if implemented across the firm?

The Printer Case Discussion

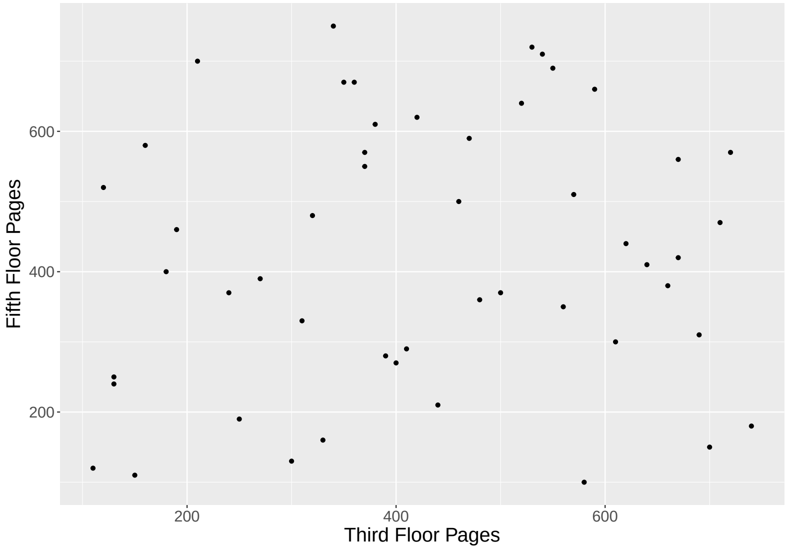
Go.

Printer Case: Numerical Summary

```r
printer <- read.csv("printer.csv") %>% tbl_df
summary(printer)
```

```
      Day             Fifth            Third      
 Min.   : 1.00   Min.   :100.0   Min.   :110.0  
 1st Qu.:13.25   1st Qu.:282.5   1st Qu.:302.5  
 Median :25.50   Median :415.0   Median :415.0  
 Mean   :25.50   Mean   :426.2   Mean   :428.2  
 3rd Qu.:37.75   3rd Qu.:577.5   3rd Qu.:577.5  
 Max.   :50.00   Max.   :750.0   Max.   :740.0  
```

Printer Case: Scatterplot ($r = 0.11$)



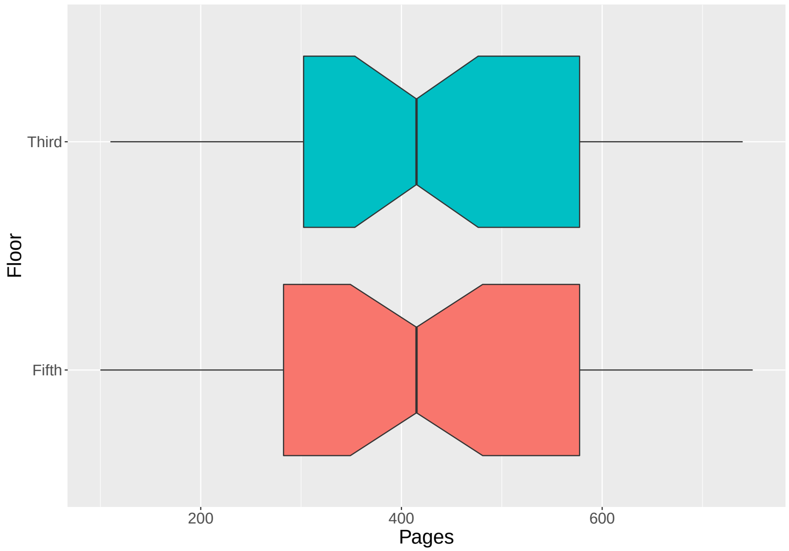
Printer Case: Gather the Columns

First, we'll gather up the data so that we can plot it more easily.

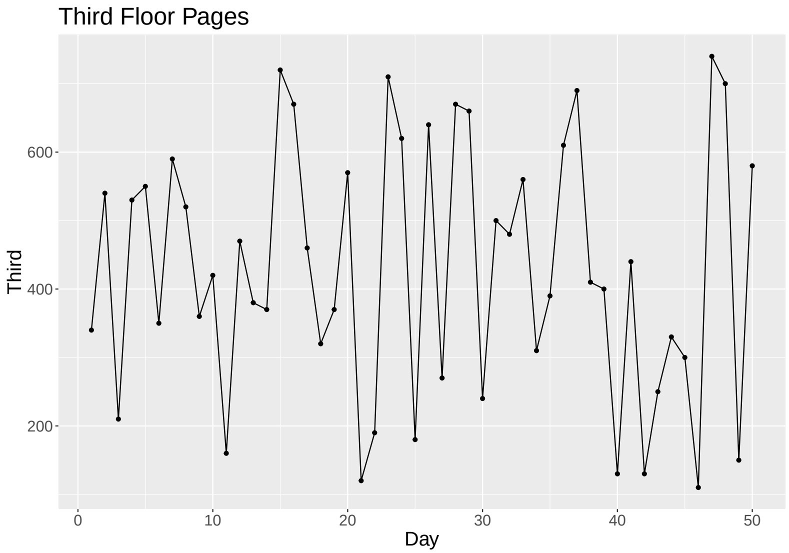
```
printer2 <- tidyr::gather(printer, Floor, Pages, -Day)
printer2
```

```
# A tibble: 100 x 3
     Day Floor Pages
   <int> <chr> <int>
 1     1 Fifth   750
 2     2 Fifth   710
 3     3 Fifth   700
 4     4 Fifth   720
 5     5 Fifth   690
 6     6 Fifth   670
 7     7 Fifth   660
 8     8 Fifth   640
 9     9 Fifth   670
```

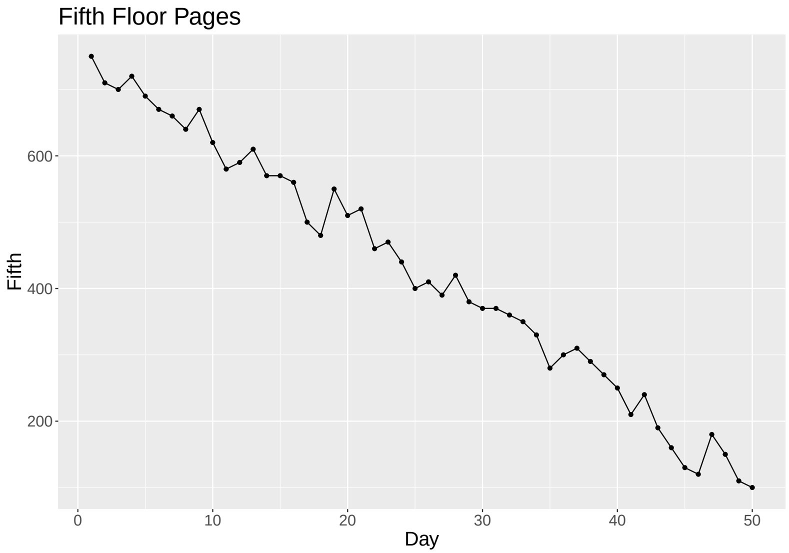
Printer Case: Comparison Boxplot



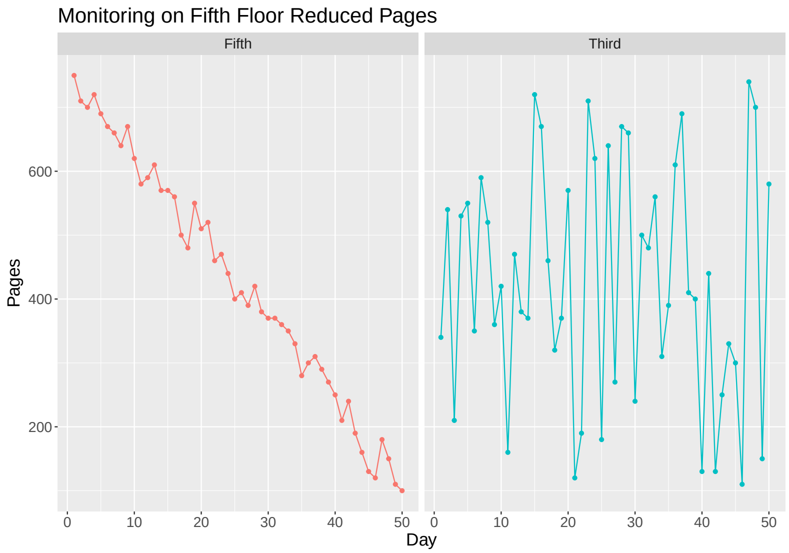
Printer Case: Third Floor



Printer Case: Fifth Floor



Comparing the Patterns over Time



Back to WCGS: Select variables, sample 500 subjects

```r
set.seed(43101)
wcgs1 <-
  wcgs %>%
  select(id, age, chol, arcus, dibpat, bmi,
         wghtcat, smoke, ncigs, chd69) %>%
  sample_n(500, replace = FALSE)
```

Dealing with Missing Data

1. Missing Completely at Random (MCAR)
   - The missing data are just a random subset of the data.
2. Missing at Random (MAR)
   - MAR means that the probability of a data point being missing has nothing to do with the missing value that would have been observed, but does have something to do with the values of some other variable that you did observe.
   - Multiple Imputation assumes the missingness is MCAR or MAR.
3. Missing not at Random ("non-ignorable" missingness)
   - Here, there is a relationship between the probability that a value is missing and what the actual (missing) value is.

What should we do about missing values?

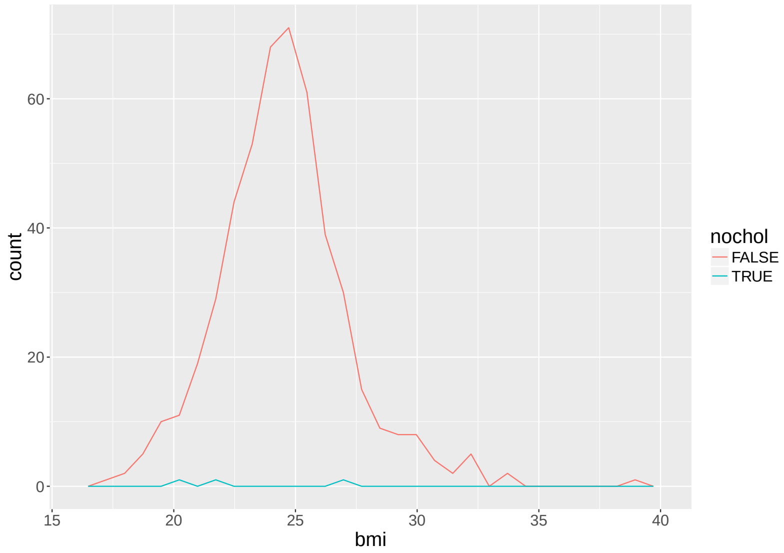
1. `ggplot2` doesn't include missing values in the plot, but it does warn that they've been removed.
2. At times you will want to try to understand what makes observations with missing values different from observations with meaningful recorded values, especially if we're thinking that the missing mechanism is MCAR or MAR.
    - We might, for instance, compare the BMI values or perhaps the smoking status for those with and without missing cholesterol values, using the `is.na()` function to make a new variable to indicate those subjects without a cholesterol level.

Do NA cholesterol look unusual in terms of BMI?

This code builds a new (logical) variable (TRUE/FALSE) to indicate a missing cholesterol level, and then we'll plot the BMI distributions for each level of the new variable.

```r
wcgs1 %>%
  mutate(
    nochol = is.na(chol)
  ) %>%
  ggplot(aes(x = bmi)) +
  geom_freqpoly(aes(col = nochol), bins = 30) +
  theme(text = element_text(size = 18))
```

Do the BMIs of people without chol look different?



Are the people without a cholesterol value unusual in terms of their smoking status?

```r
temp1 <- table(is.na(wcgs1$chol), wcgs1$smoke)
knitr::kable(temp1)
```

	No	Yes
FALSE	253	244
TRUE	2	1

What should we do about missing values?

3. How many missing values are there?
    - If the missing values are more than, say, 5% of a variable, we're going to need some strong, almost heroic assumptions in order to feel confident about using such a variable in building a model or making an inference.
    - If the amount of missing data is very small relative to the size of the data as a whole, then leaving out a few samples and just running models or comparisons ignoring those observations may not be too damaging.
    - Depending on the situation, you may want to look for other fixes besides just dropping these cases and wiping out potentially useful data.

- Some of this material comes from `this R-bloggers post`

What should we do about missing values?

Could we **impute** missing values?

- One approach is *simple* imputation, where a single value is created to "fill in" the missing observation. This is pretty easy to do, but very rarely a good idea.
  - Rarely, substituting the mean is a reasonable thing to do, as it reduces variance in your estimate of the distribution, among other problems.
  - Sometimes, but still pretty rarely, substituting in a random value observed in the rest of the data set is a reasonable thing to do.
  - Better, although still problematic, imputation approaches use other variables in the data set to predict the missing value, and contain a random component. Using other variables preserves the relationships among variables in the imputations. The random component is important so that all missing values of a single variable are not all exactly equal. One example would be to use a regression equation to predict missing values, then add a random error term.
- See http://www.theanalysisfactor.com/multiple-imputation-in-a-nutshell/

What's so bad about simple imputation?

Although there are several simple imputation approaches that solve many of the problems inherent in mean imputation, one problem remains. Because the imputed value is an estimate - a predicted value - there is uncertainty about its true value. Every statistic has uncertainty, measured by its standard error. Statistics computed using imputed data have even more uncertainty than its standard error measures. Your statistical package cannot distinguish between an imputed value and a real value.

Since the standard errors of statistics based on imputed values, such as sample means or regression coefficients, are too small, corresponding reported p-values are also too small. P-values that are reported as smaller than they, in reality, are, lead to Type I errors.

- It turns out that *multiple* imputation is a much better approach.
- Various types of "hot deck" procedures can help, too. See the `HotDeckImputation` package in R, or `this link`

So what is multiple imputation?

Multiple imputation has solved this problem by incorporating the uncertainty inherent in imputation. It has four steps:

1. Create *m* sets of imputations for the missing values using an imputation process with a random component.
2. The result is *m* full data sets. Each data set will have slightly different values for the imputed data because of the random component.
3. Analyze each completed data set. Each set of parameter estimates will differ slightly because the data differs slightly.
4. Combine results, calculating the variation in parameter estimates.

Multiple Imputation is amazing

Remarkably, *m*, the number of sufficient imputations, can be only 5 to 10 imputations, although it depends on the percentage of data that are missing. The result is unbiased parameter estimates and a full sample size, when done well.

Doing multiple imputation well, however, is not always quick or easy. First, it requires that the missing data be ignorable. Second, it requires a very good imputation model. Creating a good imputation model requires knowing your data very well and having variables that will predict missing values.

Source:
http://www.theanalysisfactor.com/multiple-imputation-in-a-nutshell/

What will we do in 431?

- Often, we'll be willing to simply exclude the data with missing values from our graphs or other analyses.
- Sometimes, we'll be willing to assume (heroically) that the data are missing at random and we'll use a simple imputation approach, via the `mice` package.
- Later in the term (and definitely in 432) we'll move on up to multiple imputation, using `mice` sometimes and `Hmisc` at other times.

Using mice to build imputations for chol

```
md.pattern(wcgs1)
```

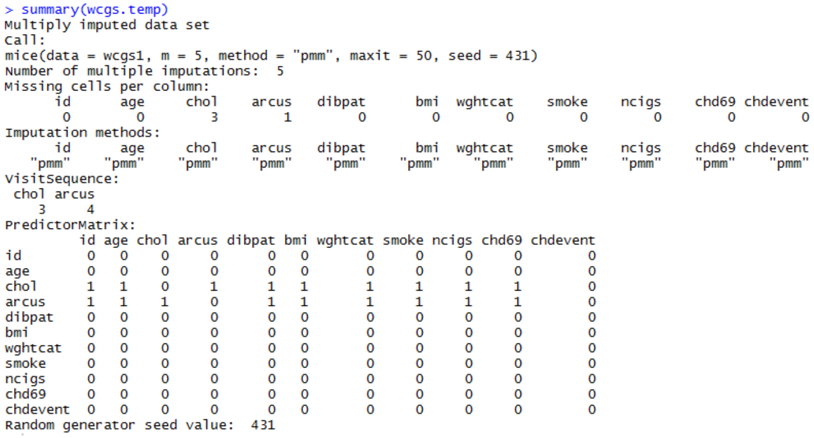
```
    id age dibpat bmi wghtcat smoke ncigs chd69 arcus
496  1   1      1   1       1     1     1     1     1
  3  1   1      1   1       1     1     1     1     1
  1  1   1      1   1       1     1     1     1     0
     0   0      0   0       0     0     0     0     1
    chol
496    1 0
  3    0 1
  1    1 1
       3 4
```

Build 5 actual imputations using the "predictive mean matching" (pmm) approach

```r
wcgs.temp <- mice(wcgs1,m=5,maxit=50,meth='pmm',seed=431)
```

```
 iter imp variable
  1   1  chol  arcus
  1   2  chol  arcus
  1   3  chol  arcus
  1   4  chol  arcus
  1   5  chol  arcus
  2   1  chol  arcus
  2   2  chol  arcus
  2   3  chol  arcus
  2   4  chol  arcus
  2   5  chol  arcus
  3   1  chol  arcus
```

View imputation results, summarized



```
> summary(wcgs.temp)
Multiply imputed data set
Call:
mice(data = wcgs1, m = 5, method = "pmm", maxit = 50, seed = 431)
Number of multiple imputations:  5
Missing cells per column:
      id      age     chol    arcus   dibpat      bmi  wghtcat    smoke    ncigs    chd69 chdevent
       0        0        3        1        0        0        0        0        0        0        0
Imputation methods:
      id      age     chol    arcus   dibpat      bmi  wghtcat    smoke    ncigs    chd69 chdevent
   "pmm"    "pmm"    "pmm"    "pmm"    "pmm"    "pmm"    "pmm"    "pmm"    "pmm"    "pmm"    "pmm"
VisitSequence:
 chol arcus
    3     4
PredictorMatrix:
         id age chol arcus dibpat bmi wghtcat smoke ncigs chd69 chdevent
id        0   0    0     0      0   0       0     0     0     0        0
age       0   0    0     0      0   0       0     0     0     0        0
chol      1   1    0     1      1   1       1     1     1     1        0
arcus     1   1    1     0      1   1       1     1     1     1        0
dibpat    0   0    0     0      0   0       0     0     0     0        0
bmi       0   0    0     0      0   0       0     0     0     0        0
wghtcat   0   0    0     0      0   0       0     0     0     0        0
smoke     0   0    0     0      0   0       0     0     0     0        0
ncigs     0   0    0     0      0   0       0     0     0     0        0
chd69     0   0    0     0      0   0       0     0     0     0        0
chdevent  0   0    0     0      0   0       0     0     0     0        0
Random generator seed value:  431
```

Inspect the imputed values, if you like

```r
wcgs.temp$imp$chol
```

```
      1   2   3   4   5
62  128 224 194 178 234
237 235 245 202 198 226
472 208 227 210 242 212
```

```r
wcgs.temp$imp$arcus
```

```
    1 2 3 4 5
480 0 0 0 0 0
```

Simple Imputation: Complete data with, let's say, the fourth of the five imputations we built

```
completedwcgs <- mice::complete(wcgs.temp,4)
```

favstats with and without imputation

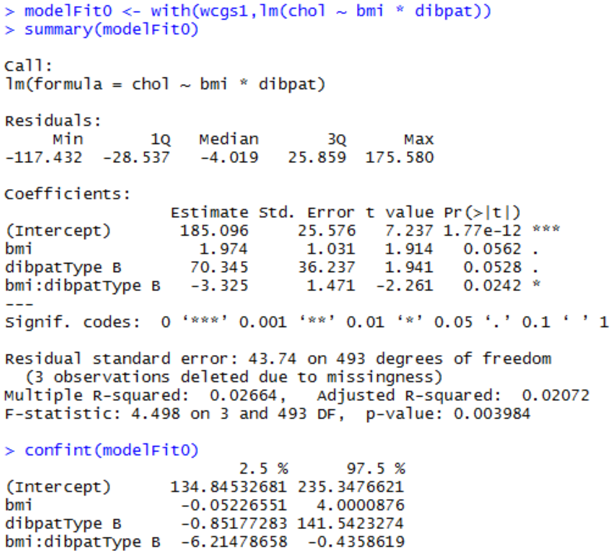
```
mosaic::favstats(wcgs1$chol)
```

```
 min   Q1 median   Q3 max    mean       sd   n missing
 110 201    224  255 400 228.167 44.20081 497       3
```

```
mosaic::favstats(completedwcgs$chol)
```

```
 min     Q1 median   Q3 max    mean       sd   n
 110 200.75    224  255 400 228.034 44.14974 500
 missing
       0
```

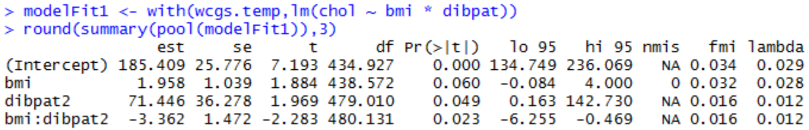
Build a Linear Model without imputation



```
> modelFit0 <- with(wcgs1,lm(chol ~ bmi * dibpat))
> summary(modelFit0)

Call:
lm(formula = chol ~ bmi * dibpat)

Residuals:
     Min       1Q   Median       3Q      Max 
-117.432  -28.537   -4.019   25.859  175.580 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)       185.096     25.576   7.237 1.77e-12 ***
bmi                 1.974      1.031   1.914   0.0562 .  
dibpatType B       70.345     36.237   1.941   0.0528 .  
bmi:dibpatType B   -3.325      1.471  -2.261   0.0242 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 43.74 on 493 degrees of freedom
  (3 observations deleted due to missingness)
Multiple R-squared:  0.02664,	Adjusted R-squared:  0.02072 
F-statistic: 4.498 on 3 and 493 DF,  p-value: 0.003984

> confint(modelFit0)
                        2.5 %      97.5 %
(Intercept)      134.84532681 235.3476621
bmi               -0.05226551   4.0000876
dibpatType B      -0.85177283 141.5423274
bmi:dibpatType B  -6.21478658  -0.4358619
```

Multiple imputation and pooling

Suppose that the next step in our analysis is to fit a linear model to the data. You may ask what imputed data set to choose. The `mice` package makes it again very easy to fit a a model to each of the imputed data sets and then pool the results together



```
> modelFit1 <- with(wcgs.temp,lm(chol ~ bmi * dibpat))
> round(summary(pool(modelFit1)),3)
                est     se      t      df Pr(>|t|)   lo 95   hi 95 nmis   fmi lambda
(Intercept) 185.409 25.776  7.193 434.927    0.000 134.749 236.069   NA 0.034  0.029
bmi           1.958  1.039  1.884 438.572    0.060  -0.084   4.000    0 0.032  0.028
dibpat2      71.446 36.278  1.969 479.010    0.049   0.163 142.730   NA 0.016  0.012
bmi:dibpat2  -3.362  1.472 -2.283 480.131    0.023  -6.255  -0.469   NA 0.016  0.012
```

Details of linear model after pooling

`modelFit1` contains the results of the fitting performed over the imputed data sets, while the `pool()` function pools them all together.

- `fmi` = fraction of missing information
- `lambda` = proportion of total variance attributable to the missing data
- Note that if we were looking at a strict alpha of 0.05, we'd have a significant `dibpat2` main effect now, when we didn't before.

Link to the Quiz

will be provided by 3 PM Thursday 2017-10-05.