

432 Homework 1 Answer Sketch

432 TAs

Due 2020-01-28. Version: 2020-01-27

Contents

Setup and Data Ingest	1
1 Question 1 (50 points)	2
1.1 Creating the BMI and BMI Category variables	2
1.2 Filtering down to what we want	3
1.3 Checking for Missing Values	3
1.4 First Attempt at Table 1	4
1.5 Making Decisions about Summary Approaches	4
1.6 Final Table 1 and Notes	7
2 Question 2. (25 points)	9
2.1 We'll start with a summary table and graph	9
2.2 A Two-Way ANOVA model with Interaction	13
2.3 A Two-Way ANOVA model without interaction (Main Effects only)	14
3 Question 3. (25 points)	15
3.1 Session Information	15

Setup and Data Ingest

Note: It's good practice not to load any more packages than you will actually use.

```
library(tableone)
library(broom)
library(naniar)
library(patchwork)
library(here)
library(janitor)
library(magrittr)
library(tidyverse)

theme_set(theme_bw())
```

Note: I loaded the data for this assignment into a subfolder of my R Project directory for Homework 1 called data. Hence, I use the following command to load in the hbp432.csv data.

```
hbp432 <- read_csv(here("data/hbp432.csv")) %>%
  clean_names() %>%
  ## could use type.convert() %>% instead of next line
  mutate_if(is.character, as.factor) %>%
  mutate(subject = as.character(subject))
```

1 Question 1 (50 points)

Note: I don't expect or need you to include the question in your response to homework assignments. I include them here to provide some context for the sketch.

Build a Table 1 to compare the subjects in practice A to the subjects in practice C on the following nine variables: age, race, Hispanic ethnicity, sex, primary insurance, body mass index, BMI category, and systolic and diastolic blood pressure. Make the Table as well as you can within R Markdown, and display the result as part of your HTML file. **Include a description of the important results from your Table 1 that does not exceed 100 words, using complete English sentences.**

1.1 Creating the BMI and BMI Category variables

With the metric system, the formula for BMI is weight (in kilograms) divided by the square of height (in meters.)

```
hbp432 <- hbp432 %>%  
  
  # first we'll create the bmi values  
  
  mutate( bmi = weight / (height*height) ) %>%  
  
  # next we'll create the bmi categories with case_when  
  # note that values of NA in bmi return NA in bmi_cat, too  
  
  mutate( bmi_cat = case_when(  
    bmi < 18.5 ~ "Underweight",  
    bmi < 25 ~ "Normal",  
    bmi < 30 ~ "Overweight",  
    bmi >= 30 ~ "Obese")) %>%  
  
  # finally we arrange the bmi_cat by the median(bmi) levels  
  # we could instead have arranged by hand using fct_relevel  
  
  mutate( bmi_cat = fct_reorder( bmi_cat, bmi, median ))
```

As a sanity check, let's ensure that the subjects in each `bmi_cat` have `bmi` values in the range we expect...

```
hbp432 %>% group_by(bmi_cat) %>%  
  summarize(n = n(), min = min(bmi), max = max(bmi))
```

```
# A tibble: 5 x 4  
  bmi_cat      n   min   max  
  <fct>    <int> <dbl> <dbl>  
1 Underweight     7  15.1  18.2  
2 Normal        77  18.7  25.0  
3 Overweight    133  25.1  30.0  
4 Obese        213   30  61.3  
5 <NA>           2  NA    NA
```

OK. The minima and maxima match what we're looking for.

Note: In practical work, we would likely collapse together the Underweight and Normal categories, since there are so few patients in the Underweight category.

```
hbp432 <- hbp432 %>%
  mutate(bmi_cat3 = fct_collapse(bmi_cat,
    "Not_Overweight" = c("Underweight", "Normal"))
  )
```

And again, a little sanity check to ensure we've not made a mistake.

```
hbp432 %>% count(bmi_cat, bmi_cat3)
```

```
# A tibble: 5 x 3
  bmi_cat    bmi_cat3      n
  <fct>      <fct>    <int>
1 Underweight Not_Overweight    7
2 Normal      Not_Overweight   77
3 Overweight  Overweight    133
4 Obese       Obese        213
5 <NA>        <NA>          2
```

1.2 Filtering down to what we want

Now that we've taken care of the BMI variables, let's filter out the subjects in practice B and D, then we can select our variables of interest. `fct_drop` is used to remove the now empty levels "B" and "D" from practice.

```
hbp432_t1 <- hbp432 %>%
  filter(practice %in% c("A", "C")) %>%
  select(subject, practice, age, race, eth_hisp, sex, insurance, bmi, bmi_cat, sbp, dbp) %>%
  mutate(practice = fct_drop(practice))
```

1.3 Checking for Missing Values

```
miss_var_summary(hbp432_t1)
```

```
# A tibble: 11 x 3
  variable n_miss pct_miss
  <chr>    <int>    <dbl>
1 eth_hisp     3    1.49
2 race         2    0.990
3 bmi          1    0.495
4 bmi_cat      1    0.495
5 subject      0     0
6 practice     0     0
7 age          0     0
8 sex          0     0
9 insurance    0     0
10 sbp         0     0
11 dbp         0     0
```

```
miss_case_summary(hbp432_t1)
```

```
# A tibble: 202 x 3
  case n_miss pct_miss
  <int> <int>    <dbl>
1    74     2    18.2
```

```

2    177      2    18.2
3     21      1     9.09
4     47      1     9.09
5     94      1     9.09
6      1      0      0
7      2      0      0
8      3      0      0
9      4      0      0
10     5      0      0
# ... with 192 more rows

```

We're missing the following values:

- `eth_hisp` on 4 subjects (3 in practice A), and
- `race` on 6 subjects (2 in practice A), and
- `bmi` on 2 subjects (1 in practice C), and
- `bmi_cat` on 2 subjects (1 in practice C).

To figure out, for example, that the subjects missing `eth_hisp` are in practice A, we could use:

```

hbp432_t1 %>%
  filter(!complete.cases()) %>%
  select(subject, practice, eth_hisp, race, bmi, bmi_cat)

```

```

# A tibble: 5 x 6
  subject practice eth_hisp race      bmi bmi_cat
  <chr>    <fct>    <fct>    <fct>    <dbl> <fct>
1 A021    A        No        <NA>    20.6 Normal
2 A047    A        <NA>      Black/AA 30.9 Obese
3 A074    A        <NA>      <NA>    35.9 Obese
4 A094    A        <NA>      Multiracial 26.8 Overweight
5 C061    C        No        Multiracial NA <NA>

```

1.4 First Attempt at Table 1

```

hw1_table1 <- CreateTableOne(data = hbp432_t1,
  vars = c("age", "race", "eth_hisp", "sex",
    "insurance", "bmi", "bmi_cat",
    "sbp", "dbp"),
  strata = c("practice"))

```

1.5 Making Decisions about Summary Approaches

Note that if we look at a `summary` of this Table 1, we see that:

- the difference in p values between the “normal” and “non-normal” versions of each of the continuous variables is small enough that if we’re going to compare the p values to, say, 0.05, it won’t matter which we choose, and
- the practical impact of the choice of p values between the “approximate” and “exact” versions of each of the categorical variables is also small.

```
summary(hw1_table1)
```

```
### Summary of continuous variables ###
```

```

practice: A
      n miss p.miss mean sd median p25 p75 min max skew kurt
age 116    0      0   71 11    70  63  78  44  89 -0.2 -0.7
bmi 116    0      0   29  6    29  25  34  17  44  0.3 -0.3
sbp 116    0      0  130 16   130 122 136  95 200  1.1  3.8
dbp 116    0      0   73 10    74  66  80  48 100 -0.2  0.2

```

```

-----
practice: C
      n miss p.miss mean sd median p25 p75 min max skew kurt
age  86     0      0   54 12    55  47  61  25  83 -0.2 -0.10
bmi  86     1      1   34  8    32  28  40  17  59  0.5  0.02
sbp  86     0      0  134 18   132 122 144  94 198  0.7  1.54
dbp  86     0      0   80 10    80  76  84  54 136  1.9 11.67

```

```

p-values
      pNormal    pNonNormal
age 2.398779e-20 2.716435e-17
bmi 2.684146e-06 4.935841e-05
sbp 1.313165e-01 1.635375e-01
dbp 4.260954e-07 1.371942e-07

```

```

Standardize mean differences
      1 vs 2
age 1.4672518
bmi 0.6733769
sbp 0.2144624
dbp 0.7443460

```

```

=====
### Summary of categorical variables ###

```

```

practice: A
      var    n miss p.miss      level freq percent cum.percent
      race 116    2   1.7   Asian/PI    2    1.8        1.8
                        Black/AA   92   80.7        82.5
                        Multiracial  4    3.5        86.0
                        White     16   14.0       100.0

      eth_hisp 116    3   2.6        No 110   97.3        97.3
                        Yes    3    2.7       100.0

      sex 116     0   0.0          F  76   65.5        65.5
                        M   40   34.5       100.0

      insurance 116    0   0.0 Commercial 39   33.6        33.6
                        Medicaid    0    0.0        33.6
                        Medicare   77   66.4       100.0
                        Uninsured   0    0.0       100.0

      bmi_cat 116    0   0.0 Underweight  2    1.7        1.7
                        Normal   26   22.4        24.1
                        Overweight 42   36.2       60.3

```

Obese	46	39.7	100.0
-------	----	------	-------

practice: C

var	n	miss	p.miss	level	freq	percent	cum.percent
race	86	0	0.0	Asian/PI	3	3.5	3.5
				Black/AA	12	14.0	17.4
				Multiracial	1	1.2	18.6
				White	70	81.4	100.0
eth_hisp	86	0	0.0	No	65	75.6	75.6
				Yes	21	24.4	100.0
sex	86	0	0.0	F	39	45.3	45.3
				M	47	54.7	100.0
insurance	86	0	0.0	Commercial	6	7.0	7.0
				Medicaid	35	40.7	47.7
				Medicare	32	37.2	84.9
				Uninsured	13	15.1	100.0
bmi_cat	86	1	1.2	Underweight	1	1.2	1.2
				Normal	5	5.9	7.1
				Overweight	27	31.8	38.8
				Obese	52	61.2	100.0

p-values

	pApprox	pExact
race	1.518397e-20	1.161424e-23
eth_hisp	8.572195e-06	2.966939e-06
sex	6.555351e-03	6.176179e-03
insurance	5.154819e-19	1.097282e-22
bmi_cat	3.292183e-03	1.426539e-03

Standardize mean differences

	1 vs 2
race	1.9286204
eth_hisp	0.6709999
sex	0.4143890
insurance	1.6588777
bmi_cat	0.5621832

1.5.1 A closer look at Age

As an example, let's plot the `age` data within each practice to evaluate Normality. Just for fun, I'll run

- a boxplot
- a histogram, and
- a normal Q-Q plot

of the ages within each practice.

```
p1 <- ggplot(hbp432_t1, aes(x = practice, y = age)) +  
  geom_boxplot(fill = c("darkorange", "green")) +
```

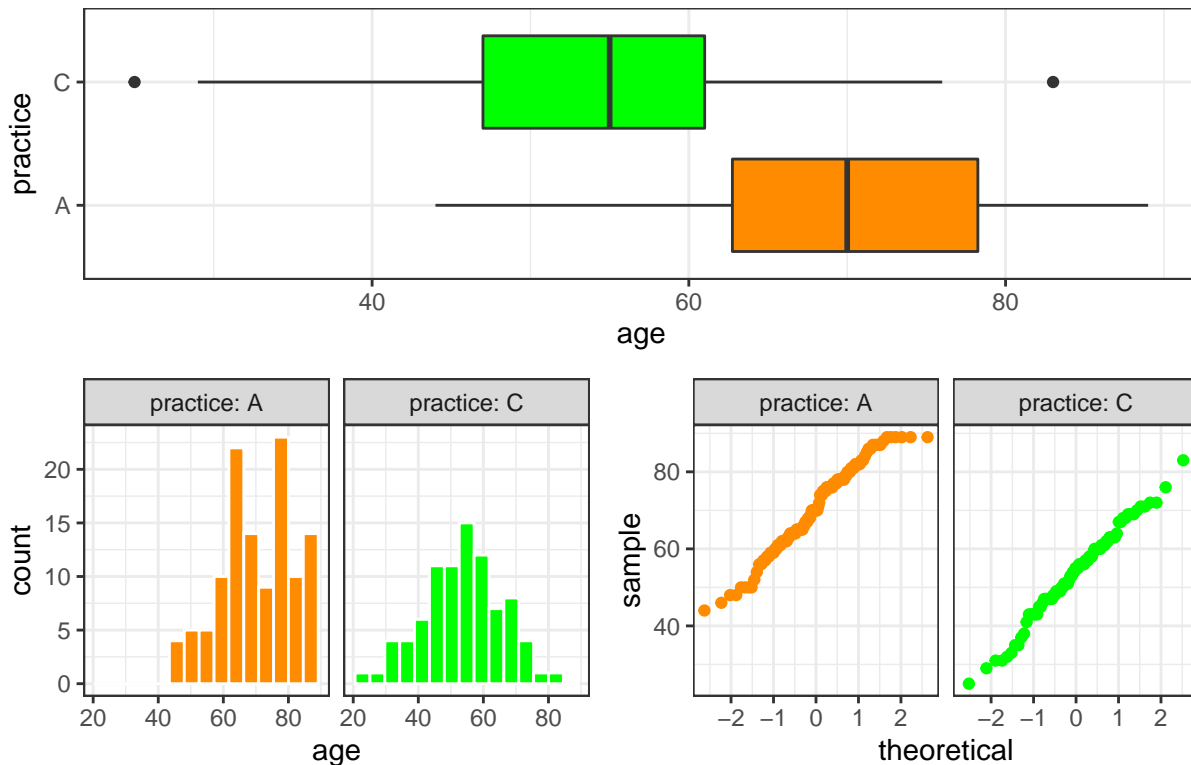
```
coord_flip()

p2 <- ggplot(hbp432_t1, aes(x = age, fill = practice)) +
  geom_histogram(bins = 15, col = "white") +
  scale_fill_manual(values = c("darkorange", "green")) +
  guides(fill = FALSE) +
  facet_wrap( ~ practice, labeller = label_both)

p3 <- ggplot(hbp432_t1, aes(sample = age, color = practice)) +
  geom_qq() +
  scale_color_manual(values = c("darkorange", "green")) +
  guides(color = FALSE) +
  facet_wrap( ~ practice, labeller = label_both)

p1 / (p2 + p3) +
  plot_annotation(title = "Comparing Age in Practices A and C")
```

Comparing Age in Practices A and C



The data aren't *perfectly* Normally distributed, of course, especially with the slight left skew in practice A, but I don't see a big problem with summarizing via means and standard deviations. We could do something similar with each of the other quantitative variables.

1.6 Final Table 1 and Notes

```
print(hw1_table1)
```

	Stratified by practice		p	test
	A	C		
n	116	86		
age (mean (SD))	70.53 (11.21)	53.79 (11.61)	<0.001	
race (%)			<0.001	
Asian/PI	2 (1.8)	3 (3.5)		
Black/AA	92 (80.7)	12 (14.0)		
Multiracial	4 (3.5)	1 (1.2)		
White	16 (14.0)	70 (81.4)		
eth_hisp = Yes (%)	3 (2.7)	21 (24.4)	<0.001	
sex = M (%)	40 (34.5)	47 (54.7)	0.007	
insurance (%)			<0.001	
Commercial	39 (33.6)	6 (7.0)		
Medicaid	0 (0.0)	35 (40.7)		
Medicare	77 (66.4)	32 (37.2)		
Uninsured	0 (0.0)	13 (15.1)		
bmi (mean (SD))	29.27 (5.90)	34.03 (8.10)	<0.001	
bmi_cat (%)			0.003	
Underweight	2 (1.7)	1 (1.2)		
Normal	26 (22.4)	5 (5.9)		
Overweight	42 (36.2)	27 (31.8)		
Obese	46 (39.7)	52 (61.2)		
sbp (mean (SD))	130.17 (16.32)	133.80 (17.51)	0.131	
dbp (mean (SD))	72.71 (9.87)	80.05 (9.85)	<0.001	

Notes for Table 1:

1. There are 3 subjects missing hispanic ethnicity (**eth_hisp**) status in practice A.
2. There are 2 subjects missing **race** in practice A.
3. There is 1 subject missing **bmi** and **bmi_cat** in practice C.
4. Results are shown in terms of means and standard deviations for quantitative variables, and t tests are used for comparisons.
5. For categorical variables, we display counts and percentages, and use Pearson chi-square tests of significance.

1.6.1 Describing the Table in a few sentences

The key conclusions of this Table 1 are that the two practices are not especially comparable on anything except for perhaps systolic blood pressure. Subjects seen in Practice A are more frequently of Black race, less often of Hispanic ethnicity, more likely to have Commercial or Medicare insurance (but less likely to have Medicaid or be uninsured) and display smaller diastolic blood pressure values than Practice C subjects. Practice A patients are also a bit older on average, more likely to be male, and have smaller BMI values.

2 Question 2. (25 points)

Now, look at the complete data, describing practices A, B, C and D. Does which **insurance** status a person has seem to have a meaningful impact on their **systolic blood pressure**, adjusting for whether or not they are on a **blood pressure medication**? Decide whether your model should include an interaction term sensibly, and then fit your choice of model and interpret and display the coefficients and other findings carefully. Be sure to provide a written explanation of your findings, in complete sentences. Responses without graphs are not complete.

2.1 We'll start with a summary table and graph

We want to understand something about the impact of insurance status on SBP, adjusting for BP medication status.

Let's start with some numbers. We'll run a count of the number of patients within the different insurance groups and medication groups, along with some basic summaries of the SBP data within each group.

Unfortunately, we'll see some NA's for the Medicare and BP medication group summaries for SBP.

```
hbp432_summaries <- hbp432 %>%
  group_by(insurance, bpmed) %>%
  summarize(n = n(), missing = n_miss(sbp),
            median = median(sbp), mean = mean(sbp), stdev = sd(sbp))

hbp432_summaries
```

```
# A tibble: 8 x 7
# Groups:   insurance [4]
  insurance bpmed    n missing median mean stdev
  <fct>     <dbl> <int>   <int>   <dbl> <dbl> <dbl>
1 Commercial     0    38      0    132.  133.  17.3
2 Commercial     1    71      0    134.  135.  17.9
3 Medicaid      0    28      0    132.  133.  12.6
4 Medicaid      1    64      0    137.  140.  23.4
5 Medicare      0    47      0    131.  131.  19.4
6 Medicare      1   165      1     NA    NA   NaN
7 Uninsured      0     6      0    127.  127.   8.75
8 Uninsured      1    13      0    149.  147.   9.57
```

Let's take a closer look at what might be causing this. There's a missing SBP - who is that?

```
hbp432 %>%
  filter(insurance == "Medicare", bpmed == "1", is.na(sbp))
```

```
# A tibble: 1 x 22
  subject practice provider age race eth_hisp sex insurance income hsgrad
  <chr>   <fct>     <fct>   <dbl> <fct> <fct>   <fct> <fct>     <dbl> <dbl>
1 D022   D         D06      76 Blac~ No      M      Medicare  73300  93.1
# ... with 12 more variables: tobacco <fct>, depdiag <fct>, height <dbl>,
# weight <dbl>, ldl <dbl>, statin <dbl>, bpmed <dbl>, sbp <dbl>, dbp <dbl>,
# bmi <dbl>, bmi_cat <fct>, bmi_cat3 <fct>
```

Yep, subject D022 is missing SBP, and the way that the numerical summaries within the `summarize` function work is that any missing values will result in an NA.

The first approach we can take is do a complete case analysis, by filtering out the missing `sbp` value prior to the summaries.

```
hbp432_summaries_fix <- hbp432 %>%
  filter(!is.na(sbp)) %>%
  group_by(insurance, bpmed) %>%
  summarize(n = n(), median = median(sbp), mean = mean(sbp), stdev = sd(sbp))
```

```
hbp432_summaries_fix
```

```
# A tibble: 8 x 6
# Groups:   insurance [4]
  insurance bpmed    n median  mean stdev
  <fct>      <dbl> <int>  <dbl> <dbl> <dbl>
1 Commercial     0    38   132.  133.  17.3
2 Commercial     1    71   134.  135.  17.9
3 Medicaid       0    28   132.  133.  12.6
4 Medicaid       1    64   137.  140.  23.4
5 Medicare        0    47   131.  131.  19.4
6 Medicare        1   164   132.  135.  18.5
7 Uninsured       0     6   127.  127.   8.75
8 Uninsured       1    13   149.  147.   9.57
```

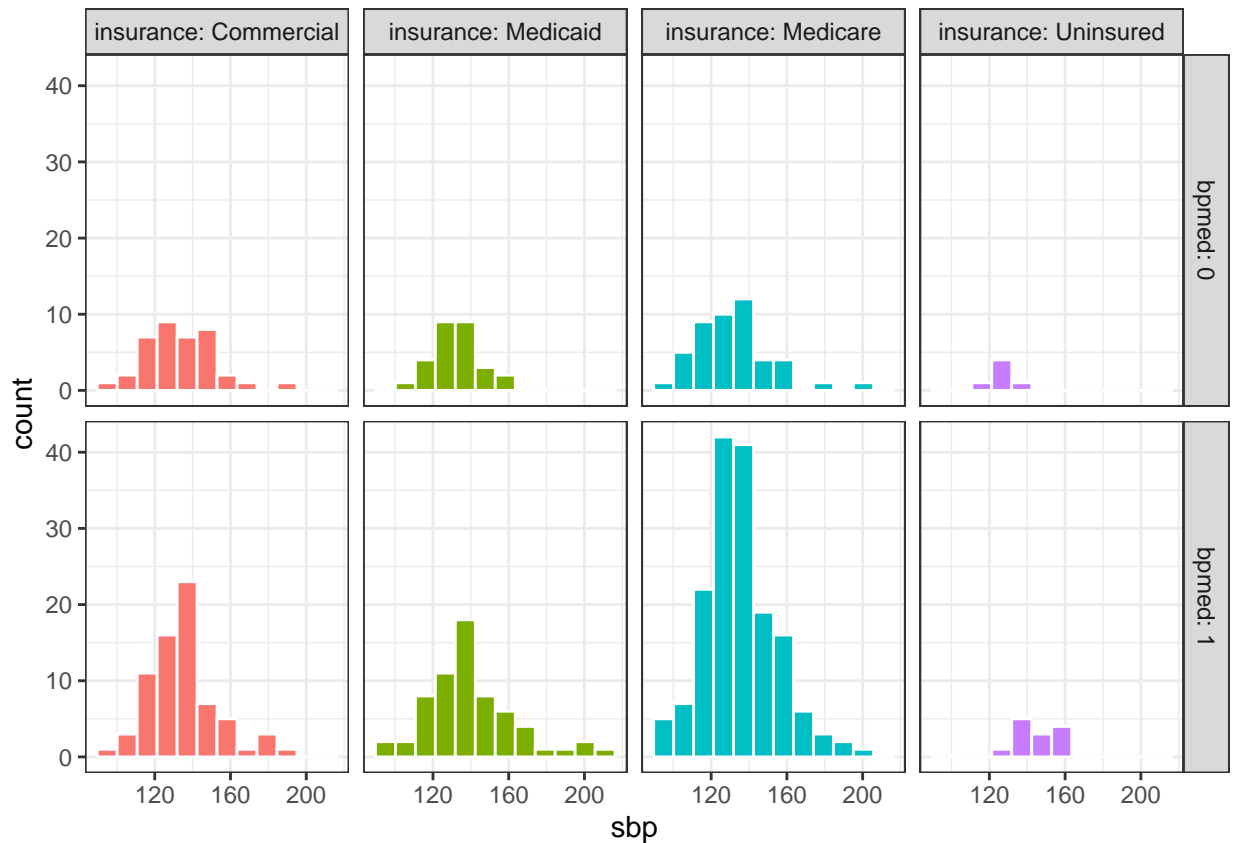
Another way to achieve a complete case analysis here is by using `na.rm = TRUE` to remove the missing values during these summary calculations.

```
hbp432 %>%
  group_by(insurance, bpmed) %>%
  summarize(n = n(), median = median(sbp, na.rm = TRUE),
            mean = mean(sbp, na.rm = TRUE), stdev = sd(sbp, na.rm = TRUE))
```

```
# A tibble: 8 x 6
# Groups:   insurance [4]
  insurance bpmed    n median  mean stdev
  <fct>      <dbl> <int>  <dbl> <dbl> <dbl>
1 Commercial     0    38   132.  133.  17.3
2 Commercial     1    71   134.  135.  17.9
3 Medicaid       0    28   132.  133.  12.6
4 Medicaid       1    64   137.  140.  23.4
5 Medicare        0    47   131.  131.  19.4
6 Medicare        1   165   132.  135.  18.5
7 Uninsured       0     6   127.  127.   8.75
8 Uninsured       1    13   149.  147.   9.57
```

We have some pretty small cell sizes, especially in the Uninsured groups, but the medians and means are quite close in each group. Let's try some graphs, first of the raw data distributions...

```
hbp432 %>%
  filter(!is.na(sbp)) %>%
  ggplot(., aes(x = sbp, fill = factor(insurance))) +
  geom_histogram(bins = 12, col = "white") +
  guides(fill = FALSE) +
  facet_grid(bpmed ~ insurance, labeller = label_both)
```

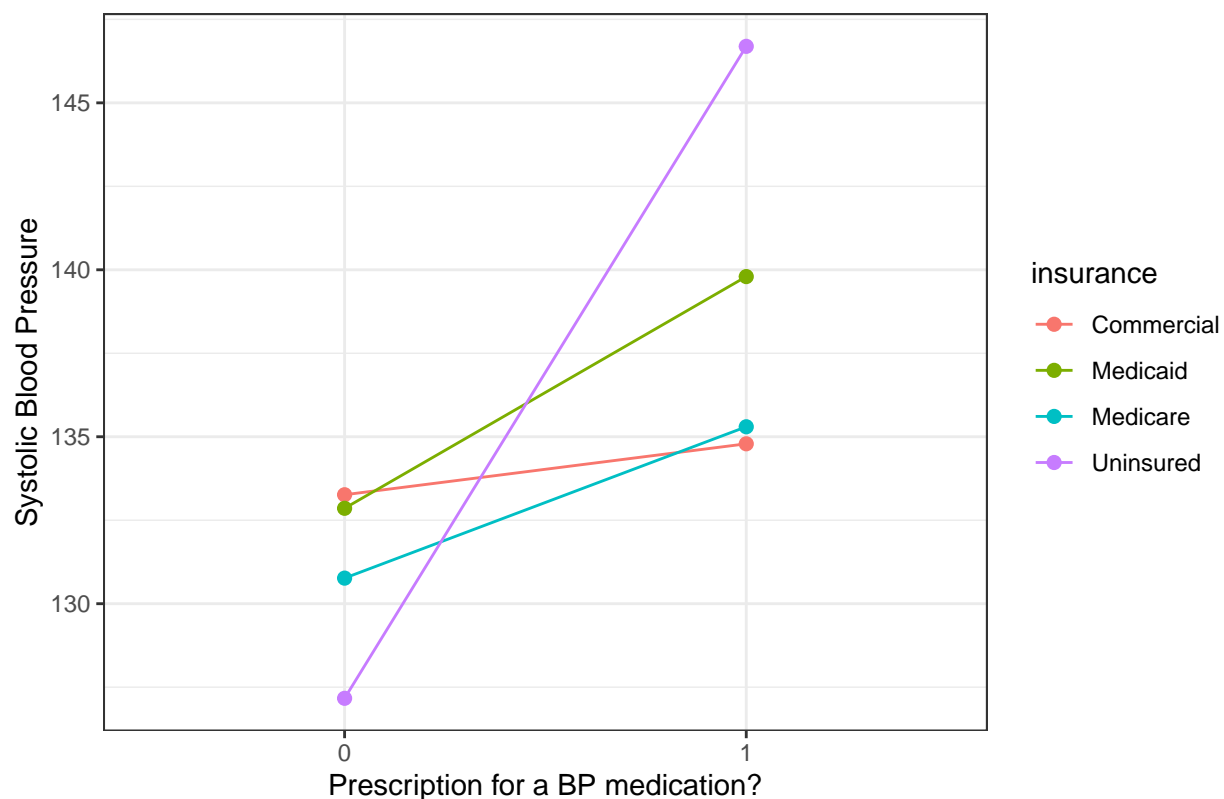


There aren't any major problems, except the small counts for the Uninsured groups making those graphs hard to interpret, so we can assume that a mean and standard deviation might be reasonable choices to summarize the data from these four groups.

So, let's try graphing the means in an interaction plot.

```
ggplot(hbp432_summaries_fix,
  aes(x = factor(bpmmed), y = mean, col = insurance)) +
  geom_point(size = 2) +
  geom_line(aes(group = insurance)) +
  labs(y = "Systolic Blood Pressure",
    x = "Prescription for a BP medication?",
    title = "Interaction Plot for SBP in the hbp432 data")
```

Interaction Plot for SBP in the hbp432 data

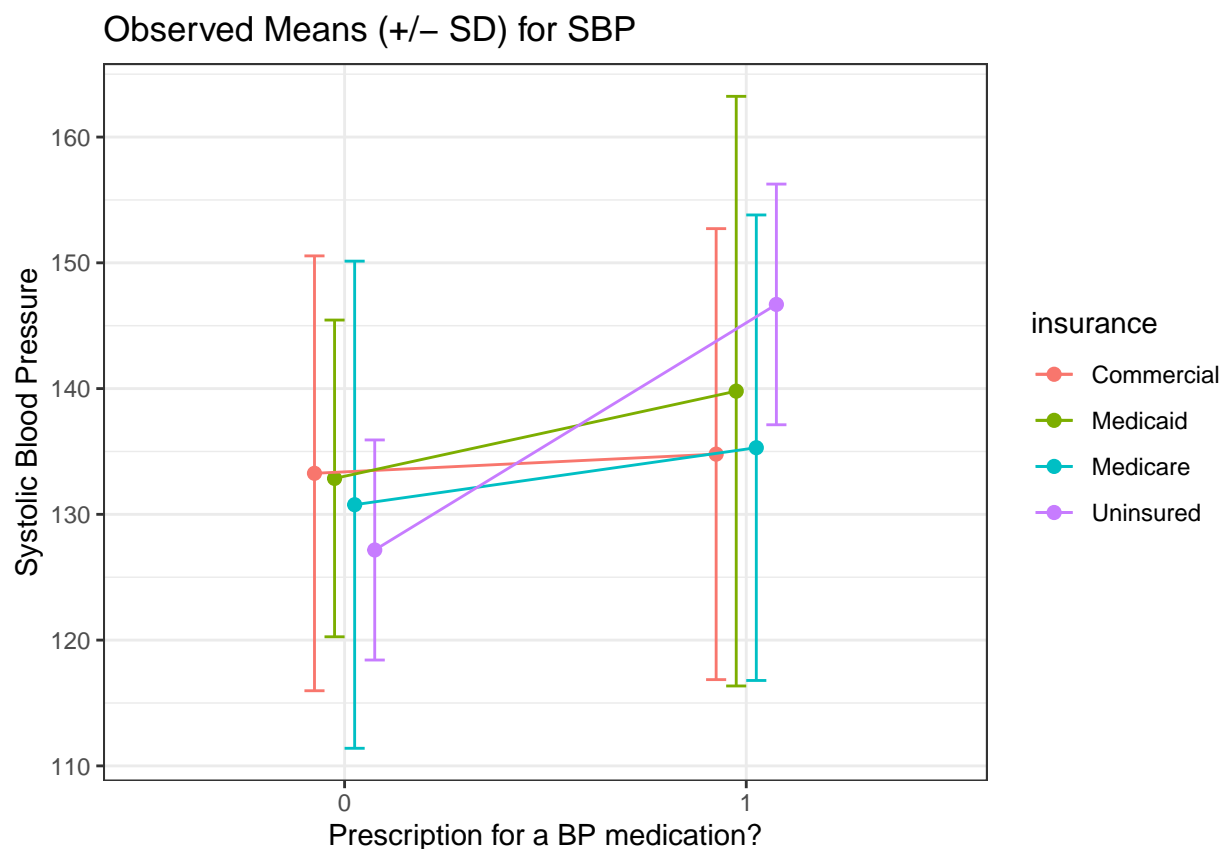


The lines joining the prescription groups vary substantially by insurance type, and are not parallel to each other. Medicaid and Medicare look fairly parallel, but the others definitely don't.

As an alternative, we could use the “dodge” approach to build an interaction plot with error bars, like this...

```
pd <- position_dodge(0.2)

ggplot(hbp432_summaries_fix, aes(x = factor(bpmed), y = mean, col = insurance)) +
  geom_errorbar(aes(ymin = mean - stdev,
                    ymax = mean + stdev),
               width = 0.2, position = pd) +
  geom_point(size = 2, position = pd) +
  geom_line(aes(group = insurance), position = pd) +
  labs(y = "Systolic Blood Pressure",
       x = "Prescription for a BP medication?",
       title = "Observed Means (+/- SD) for SBP")
```



There definitely seems to be a modest interaction between `bpmed` and `insurance`. More specifically:

- There's a small increase in SBP for Medicare and Medicaid patients with a BP medication compared to those without.
- There's a much larger increase in SBP for Uninsured patients with a BP medication compared to those without.
- And, there's not much of a change in SBP for Commercial patients with a BP medication compared to those without.

2.2 A Two-Way ANOVA model with Interaction

The previous graph indicates that there might be a meaningful interaction, so let's run that model first.

```
hw1_q2_with_int <- hbp432 %>% lm(sbp ~ insurance * bpmed)
anova(hw1_q2_with_int)
```

Analysis of Variance Table

Response: sbp

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
insurance	3	1376	458.75	1.3197	0.2674
bpmed	1	2066	2065.52	5.9419	0.0152 *
insurance:bpmed	3	1246	415.29	1.1947	0.3114
Residuals	423	147043	347.62		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```
tidy(hw1_q2_with_int, conf.int = TRUE) %>%
  select(term, estimate, std.error, p.value, conf.low, conf.high) %>%
  knitr::kable(digits = 3)
```

term	estimate	std.error	p.value	conf.low	conf.high
(Intercept)	133.263	3.025	0.000	127.318	139.208
insuranceMedicaid	-0.406	4.644	0.930	-9.533	8.721
insuranceMedicare	-2.497	4.067	0.540	-10.492	5.498
insuranceUninsured	-6.096	8.191	0.457	-22.196	10.003
bpmed	1.526	3.748	0.684	-5.841	8.892
insuranceMedicaid:bpmed	5.414	5.647	0.338	-5.686	16.514
insuranceMedicare:bpmed	3.007	4.854	0.536	-6.533	12.548
insuranceUninsured:bpmed	18.000	9.936	0.071	-1.530	37.530

The main conclusion here is the interaction is neither significant, nor does it account for a large fraction of the variation in the SBP values (0.8%). We could interpret this result, probably by specifying the predictions that the model would make for each possible combination of **insurance** and **bpmed**, but perhaps it's reasonable to assume the interaction is small enough to motivate a two-factor ANOVA model without interaction.

2.3 A Two-Way ANOVA model without interaction (Main Effects only)

```
hw1_q2_no_int <- hbp432 %$% lm(sbp ~ insurance + bpmed)
anova(hw1_q2_no_int)
```

Analysis of Variance Table

```
Response: sbp
      Df Sum Sq Mean Sq F value    Pr(>F)
insurance  3   1376   458.75    1.3179  0.26801
bpmed      1   2066  2065.52    5.9338  0.01526 *
Residuals 426 148289   348.10
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(hw1_q2_no_int)
```

Call:

```
lm(formula = sbp ~ insurance + bpmed)
```

Residuals:

```
      Min       1Q   Median       3Q      Max
-45.186 -11.455  -2.043   8.747  70.814
```

Coefficients:

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   131.0434     2.2212  58.996  <2e-16 ***
insuranceMedicaid  3.2095     2.6430   1.214   0.2253
insuranceMedicare -0.5888     2.2155  -0.266   0.7906
insuranceUninsured  6.1074     4.6388   1.317   0.1887
bpmed          4.9334     2.0253   2.436   0.0153 *
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 18.66 on 426 degrees of freedom
(1 observation deleted due to missingness)
```

```
Multiple R-squared:  0.02268,    Adjusted R-squared:  0.01351
```

```
F-statistic: 2.472 on 4 and 426 DF,  p-value: 0.04398
```

None of the different insurance status levels have a particularly strong impact on SBP, but subjects without BP medication have statistically detectably lower SBP levels, on average.

3 Question 3. (25 points)

How does the sage advice provided by George Box (and echoed by David Spiegelhalter, especially in Chapter 5 of The Art of Statistics) that - all models are wrong, but some are useful - apply to the results you have obtained in Question 2? Write an essay of 150-250 words (using complete sentences, and examples derived from your modeling) that explains how this advice is connected to your thinking about presenting your results.

We don't provide sketches for essay questions. We will share a nice example or two after grading.

3.1 Session Information

```
sessioninfo::session_info()
```

```
- Session info -----
setting  value
version  R version 3.6.2 (2019-12-12)
os       Windows 10 x64
system   x86_64, mingw32
ui       RTerm
language (EN)
collate  English_United States.1252
ctype    English_United States.1252
tz       America/New_York
date     2020-01-27

- Packages -----
package      * version    date          lib source
assertthat   0.2.1      2019-03-21   [1] CRAN (R 3.6.1)
backports    1.1.5      2019-10-02   [1] CRAN (R 3.6.1)
broom        * 0.5.3      2019-12-14   [1] CRAN (R 3.6.2)
cellranger   1.1.0      2016-07-27   [1] CRAN (R 3.6.1)
class        7.3-15     2019-01-01   [2] CRAN (R 3.6.2)
cli          2.0.1      2020-01-08   [1] CRAN (R 3.6.2)
colorspace   1.4-1      2019-03-18   [1] CRAN (R 3.6.1)
crayon       1.3.4      2017-09-16   [1] CRAN (R 3.6.1)
DBI          1.1.0      2019-12-15   [1] CRAN (R 3.6.1)
dbplyr       1.4.2      2019-06-17   [1] CRAN (R 3.6.1)
digest       0.6.23     2019-11-23   [1] CRAN (R 3.6.1)
dplyr        * 0.8.3      2019-07-04   [1] CRAN (R 3.6.1)
e1071        1.7-3      2019-11-26   [1] CRAN (R 3.6.1)
```

ellipsis	0.3.0	2019-09-20	[1]	CRAN	(R 3.6.1)
evaluate	0.14	2019-05-28	[1]	CRAN	(R 3.6.1)
fansi	0.4.1	2020-01-08	[1]	CRAN	(R 3.6.2)
farver	2.0.3	2020-01-16	[1]	CRAN	(R 3.6.2)
forcats	* 0.4.0	2019-02-17	[1]	CRAN	(R 3.6.1)
fs	1.3.1	2019-05-06	[1]	CRAN	(R 3.6.1)
generics	0.0.2	2018-11-29	[1]	CRAN	(R 3.6.1)
ggplot2	* 3.2.1	2019-08-10	[1]	CRAN	(R 3.6.1)
glue	1.3.1	2019-03-12	[1]	CRAN	(R 3.6.2)
gtable	0.3.0	2019-03-25	[1]	CRAN	(R 3.6.1)
haven	2.2.0	2019-11-08	[1]	CRAN	(R 3.6.1)
here	* 0.1	2017-05-28	[1]	CRAN	(R 3.6.2)
highr	0.8	2019-03-20	[1]	CRAN	(R 3.6.1)
hms	0.5.3	2020-01-08	[1]	CRAN	(R 3.6.2)
htmltools	0.4.0	2019-10-04	[1]	CRAN	(R 3.6.1)
httr	1.4.1	2019-08-05	[1]	CRAN	(R 3.6.1)
janitor	* 1.2.0	2019-04-21	[1]	CRAN	(R 3.6.1)
jsonlite	1.6	2018-12-07	[1]	CRAN	(R 3.6.1)
knitr	1.27	2020-01-16	[1]	CRAN	(R 3.6.2)
labeling	0.3	2014-08-23	[1]	CRAN	(R 3.6.0)
labelled	2.2.1	2019-05-26	[1]	CRAN	(R 3.6.1)
lattice	0.20-38	2018-11-04	[1]	CRAN	(R 3.6.2)
lazyeval	0.2.2	2019-03-15	[1]	CRAN	(R 3.6.1)
lifecycle	0.1.0	2019-08-01	[1]	CRAN	(R 3.6.1)
lubridate	1.7.4	2018-04-11	[1]	CRAN	(R 3.6.1)
magrittr	* 1.5	2014-11-22	[1]	CRAN	(R 3.6.2)
MASS	7.3-51.5	2019-12-20	[1]	CRAN	(R 3.6.2)
Matrix	1.2-18	2019-11-27	[2]	CRAN	(R 3.6.2)
mitools	2.4	2019-04-26	[1]	CRAN	(R 3.6.1)
modelr	0.1.5	2019-08-08	[1]	CRAN	(R 3.6.1)
munSELL	0.5.0	2018-06-12	[1]	CRAN	(R 3.6.1)
naniar	* 0.4.2	2019-02-15	[1]	CRAN	(R 3.6.1)
nlme	3.1-142	2019-11-07	[2]	CRAN	(R 3.6.2)
patchwork	* 1.0.0	2019-12-01	[1]	CRAN	(R 3.6.1)
pillar	1.4.3	2019-12-20	[1]	CRAN	(R 3.6.2)
pkgconfig	2.0.3	2019-09-22	[1]	CRAN	(R 3.6.1)
plyr	1.8.5	2019-12-10	[1]	CRAN	(R 3.6.1)
purrr	* 0.3.3	2019-10-18	[1]	CRAN	(R 3.6.1)
R6	2.4.1	2019-11-12	[1]	CRAN	(R 3.6.1)
Rcpp	1.0.3	2019-11-08	[1]	CRAN	(R 3.6.1)
readr	* 1.3.1	2018-12-21	[1]	CRAN	(R 3.6.1)
readxl	1.3.1	2019-03-13	[1]	CRAN	(R 3.6.1)
reprex	0.3.0	2019-05-16	[1]	CRAN	(R 3.6.1)
reshape2	1.4.3	2017-12-11	[1]	CRAN	(R 3.6.1)
rlang	0.4.2	2019-11-23	[1]	CRAN	(R 3.6.1)
rmarkdown	2.0	2019-12-12	[1]	CRAN	(R 3.6.2)
rprojroot	1.3-2	2018-01-03	[1]	CRAN	(R 3.6.1)
rstudioapi	0.10	2019-03-19	[1]	CRAN	(R 3.6.1)
rvest	0.3.5	2019-11-08	[1]	CRAN	(R 3.6.1)
scales	1.1.0	2019-11-18	[1]	CRAN	(R 3.6.1)
sessioninfo	1.1.1	2018-11-05	[1]	CRAN	(R 3.6.1)
snakecase	0.11.0	2019-05-25	[1]	CRAN	(R 3.6.1)
stringi	1.4.5	2020-01-11	[1]	CRAN	(R 3.6.2)
stringr	* 1.4.0	2019-02-10	[1]	CRAN	(R 3.6.1)

survey	3.36	2019-04-27	[1]	CRAN	(R 3.6.2)
survival	3.1-8	2019-12-03	[1]	CRAN	(R 3.6.2)
tableone	* 0.10.0	2019-02-17	[1]	CRAN	(R 3.6.2)
tibble	* 2.1.3	2019-06-06	[1]	CRAN	(R 3.6.1)
tidyr	* 1.0.0	2019-09-11	[1]	CRAN	(R 3.6.1)
tidyselect	0.2.5	2018-10-11	[1]	CRAN	(R 3.6.1)
tidyverse	* 1.3.0	2019-11-21	[1]	CRAN	(R 3.6.2)
utf8	1.1.4	2018-05-24	[1]	CRAN	(R 3.6.1)
vctrs	0.2.1	2019-12-17	[1]	CRAN	(R 3.6.2)
visdat	0.5.3	2019-02-15	[1]	CRAN	(R 3.6.1)
withr	2.1.2	2018-03-15	[1]	CRAN	(R 3.6.1)
xfun	0.12	2020-01-13	[1]	CRAN	(R 3.6.2)
xml2	1.2.2	2019-08-09	[1]	CRAN	(R 3.6.1)
yaml	2.2.0	2018-07-25	[1]	CRAN	(R 3.6.0)
zeallot	0.1.0	2018-01-28	[1]	CRAN	(R 3.6.1)
zoo	1.8-7	2020-01-10	[1]	CRAN	(R 3.6.2)

[1] C:/Users/Thomas/Documents/R/win-library/3.6

[2] C:/Program Files/R/R-3.6.2/library