431 Class 11

thomaselove.github.io/431

2021-09-28

Today's Agenda

1. Developing Four Models for `sbp` using `dbp` (and `insurance`)
2. Fitting a Bayesian Linear Model with default priors (`m2`)
3. Including Insurance without (`m3`) and with (`m4`) interaction with `dbp` in linear models
4. Visualizing Categorical Data
5. Assessing Association in Cross-Tabulations

Today's Packages

```
library(broom)
library(equatiomatic) # new today
library(ggrepel) # sort of new today
library(glue) # sort of new today
library(janitor)
library(knitr)
library(magrittr)
library(patchwork)
library(rstanarm) # special today
library(tidyverse)

theme_set(theme_bw())
```

Today's Data

Again, we'll use an R data set (`.Rds`) to import the `dm1000` data.

```
dm1000 <- read_rds("data/dm_1000.Rds")
```

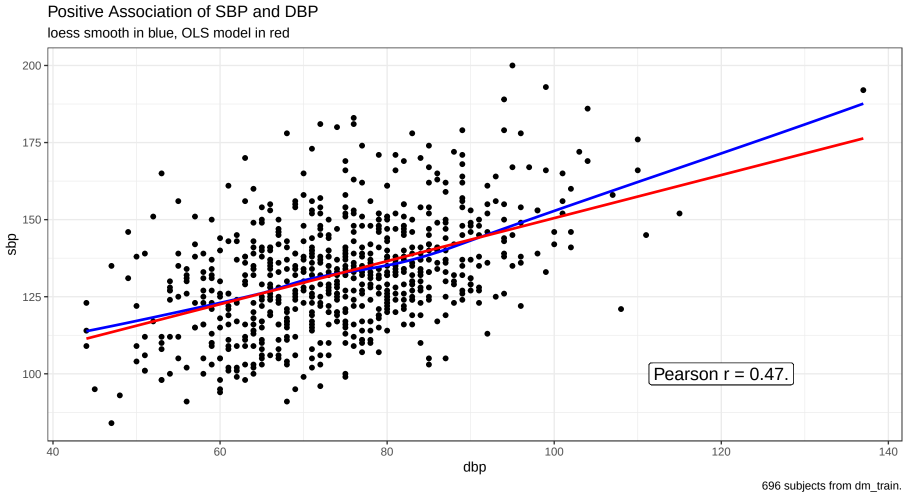
Then, we'll again partition the `dm1000` cases with complete BP data into training and test samples.

```
dm994 <- dm1000 %>% filter(complete.cases(sbp, dbp)) %>%
  select(subject, sbp, dbp, insurance)

set.seed(4312021) # for replicating the sampling later
dm_train <- dm994 %>% sample_frac(0.7)
dm_test <- dm994 %>% anti_join(dm_train, by = "subject")
```

Back to Regression: Can `dbp` predict `sbp`?

Plotting sbp vs. dbp (training set)



Model `m1` for `sbp` using `dbp` (training set)

```
m1_train <- lm(sbp ~ dbp, data = dm_train)

tidy(m1_train, conf.int = TRUE, conf.level = 0.90) %>%
  select(term, estimate, conf.low, conf.high) %>% kable()
```

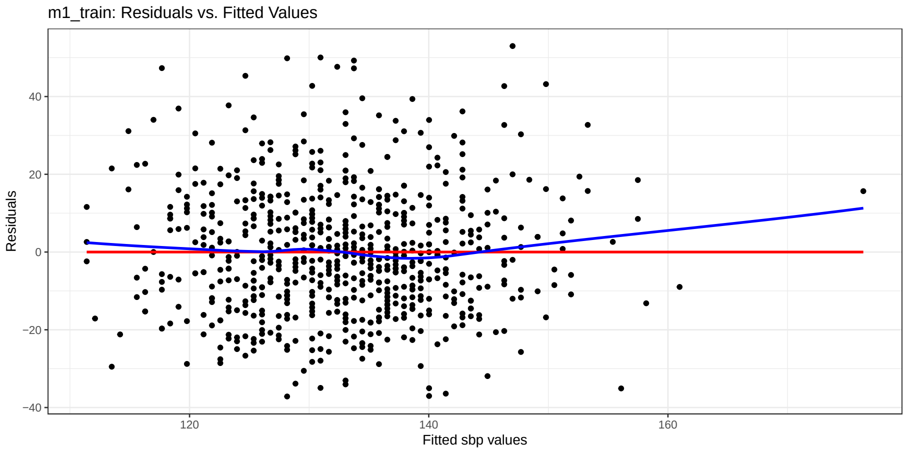
term	estimate	conf.low	conf.high
(Intercept)	80.6798905	74.4662421	86.893539
dbp	0.6982168	0.6160396	0.780394

```
glance(m1_train) %>% select(nobs, r.squared, adj.r.squared,
                            sigma, AIC, BIC) %>% kable()
```

nobs	r.squared	adj.r.squared	sigma	AIC	BIC
696	0.2200811	0.2189573	16.11605	5848.663	5862.299

m1_train: Residuals vs. Predicted (Fitted) Values

```
m1_train_aug <- augment(m1_train, data = dm_train)
```



Use model `m1_train` to predict SBP in `dm_test`

```
m1_test_aug <- augment(m1_train, newdata = dm_test)

mosaic::favstats(~ abs(.resid), data = m1_test_aug) %>%
  select(n, min, median, max, mean, sd) %>% kable(digits = 3)
```

n	min	median	max	mean	sd
298	0.028	9.822	71.066	12.139	10.177

```
sqrt(mean(m1_test_aug$.resid^2))
```

```
[1] 15.83017
```

Summary	Model `m1`
Mean Absolute Prediction Error	12.139
Maximum Absolute Prediction Error	71.066
Root Mean Squared Prediction Error (RMSPE)	15.83

Is this the only linear model R can fit to these data?

Nope.

```
library(rstanarm)

m2_train <- stan_glm(sbp ~ dbp, data = dm_train)
```

```
SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 0 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transiti
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
```

Default Prior Details

```
prior_summary(m2_train)
```

```
> prior_summary(m2_train)
Priors for model 'm2_train'
------
Intercept (after predictors centered)
  Specified prior:
    ~ normal(location = 133, scale = 2.5)
  Adjusted prior:
    ~ normal(location = 133, scale = 46)

Coefficients
  Specified prior:
    ~ normal(location = 0, scale = 2.5)
  Adjusted prior:
    ~ normal(location = 0, scale = 3.7)

Auxiliary (sigma)
  Specified prior:
    ~ exponential(rate = 1)
  Adjusted prior:
    ~ exponential(rate = 0.055)
------
See help('prior_summary.stanreg') for more details
```

Bayesian fitted linear model for our sbp data

```
print(m2_train)
```

```
stan_glm
 family:       gaussian [identity]
 formula:      sbp ~ dbp
 observations: 696
 predictors:   2
------
            Median MAD_SD
(Intercept) 80.6    4.0
dbp          0.7    0.1

Auxiliary parameter(s):
      Median MAD_SD
sigma 16.2    0.4

------
```

Is the Bayesian model (with default prior) very different from our `lm` in this situation?

```
broom::tidy(m1_train) # fit with lm
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   80.7      3.77        21.4 2.47e-78
2 dbp            0.698    0.0499      14.0 2.23e-39
```

```
broom.mixed::tidy(m2_train) # stan_glm with default priors
```

```
# A tibble: 2 x 3
  term        estimate std.error
  <chr>          <dbl>     <dbl>
1 (Intercept)   80.6      3.97
2 dbp            0.699    0.0522
```

Obtaining fits and residuals from Model m2

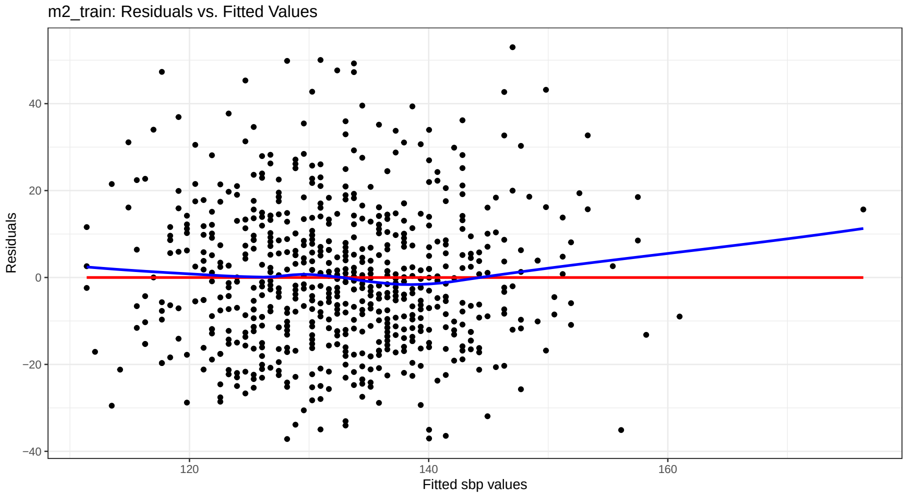
In the model training sample

```
m2_train_aug <- dm_train %>% select(subject, sbp, dbp) %>%
  mutate(.fitted = predict(m2_train, newdata = dm_train),
         .resid = sbp - .fitted)
```

In the model test sample

```
m2_test_aug <- dm_test %>% select(subject, sbp, dbp) %>%
  mutate(.fitted = predict(m2_train, newdata = dm_test),
         .resid = sbp - .fitted)
```

Residuals vs. Fitted Values from Model `m2` (training)



Out-of-Sample (Test Set) Error Summaries (m2)

```
mosaic::favstats(~ abs(.resid), data = m2_test_aug) %>%
  select(n, min, median, max, mean, sd) %>% kable(digits = 3)
```

n	min	median	max	mean	sd
298	0.036	9.824	71.059	12.14	10.177

```
sqrt(mean(m2_test_aug$.resid^2))
```

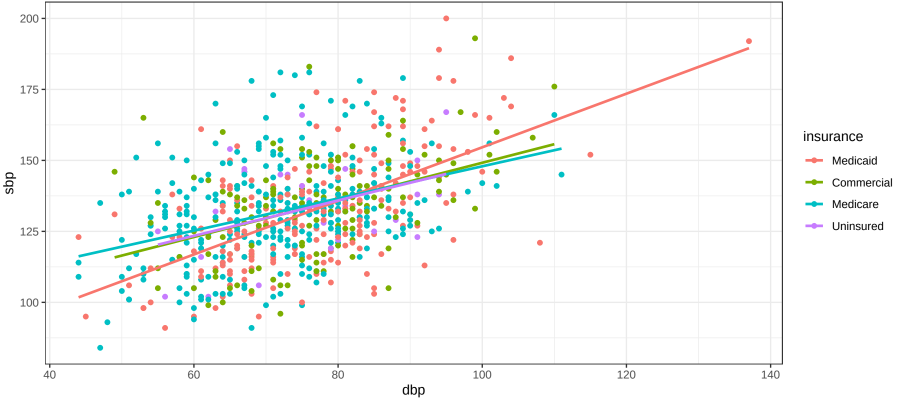
```
[1] 15.83019
```

Test Set Error Summary	OLS model m1	Bayes model m2
Mean Absolute Prediction Error	12.139	12.14
Maximum Absolute Prediction Error	71.066	71.059
Root Mean Squared Prediction Error	15.83	15.83

What if we add another predictor? (Insurance)

Plotting sbp vs. dbp and insurance

```
ggplot(data = dm_train, aes(x = dbp, y = sbp,
                 col = insurance, group = insurance)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE)
```



Two possible models

```
m3_train <- lm(sbp ~ dbp + insurance, data = dm_train)
m4_train <- lm(sbp ~ dbp * insurance, data = dm_train)
```

- What is the difference between `m3` and `m4`?
    - Model `m3` will allow the intercept term of the `sbp`-`dbp` relationship to vary depending on insurance.
    - Model `m4` will allow both the slope and intercept of the `sbp`-`dbp` relationship to vary depending on insurance.

Equation for m3 (sbp ~ dbp + insurance)

```
extract_eq(m3_train, use_coefs = TRUE,
           wrap = TRUE, terms_per_line = 2)
```

$$
\begin{aligned}
\widehat{\text{sbp}} &= 77.58 + 0.72(\text{dbp}) + \\
&\quad 1.11(\text{insurance}_{\text{Commercial}}) + 2.73(\text{insurance}_{\text{Medicare}}) + \\
&\quad 1.16(\text{insurance}_{\text{Uninsured}})
\end{aligned}
$$

- Predicted sbp by m3 for a Commercial subject?

Equation for m3 (sbp ~ dbp + insurance)

```
extract_eq(m3_train, use_coefs = TRUE,
           wrap = TRUE, terms_per_line = 2)
```

$$
\begin{aligned}
\widehat{\text{sbp}} &= 77.58 + 0.72(\text{dbp}) + \\
&\quad 1.11(\text{insurance}_{\text{Commercial}}) + 2.73(\text{insurance}_{\text{Medicare}}) + \\
&\quad 1.16(\text{insurance}_{\text{Uninsured}})
\end{aligned}
$$

- Predicted `sbp` by `m3` for a Commercial subject?
- `sbp` = 77.58 + 0.72*`dbp` + 1.11(1) + 2.73(0) + 1.16(0)

Equation for m3 (sbp ~ dbp + insurance)

```
extract_eq(m3_train, use_coefs = TRUE,
           wrap = TRUE, terms_per_line = 2)
```

$$
\begin{aligned}
\widehat{\text{sbp}} &= 77.58 + 0.72(\text{dbp}) + \\
&\quad 1.11(\text{insurance}_{\text{Commercial}}) + 2.73(\text{insurance}_{\text{Medicare}}) + \\
&\quad 1.16(\text{insurance}_{\text{Uninsured}})
\end{aligned}
$$

- Predicted sbp by m3 for a Commercial subject?
- sbp = 77.58 + 0.72*dbp + 1.11(1) + 2.73(0) + 1.16(0)
- sbp = 78.69 + 0.72*dbp

Equation for m3 (sbp ~ dbp + insurance)

```
extract_eq(m3_train, use_coefs = TRUE,
           wrap = TRUE, terms_per_line = 2)
```

$$
\begin{aligned}
\widehat{\text{sbp}} &= 77.58 + 0.72(\text{dbp}) + \\
&\quad 1.11(\text{insurance}_{\text{Commercial}}) + 2.73(\text{insurance}_{\text{Medicare}}) + \\
&\quad 1.16(\text{insurance}_{\text{Uninsured}})
\end{aligned}
$$

- Predicted `sbp` by `m3` for a Commercial subject?
- `sbp` = 77.58 + 0.72*`dbp` + 1.11(1) + 2.73(0) + 1.16(0)
- `sbp` = 78.69 + 0.72*`dbp`
- For a Medicaid subject, `m3` predicts `sbp` = 77.58 + 0.72 `dbp`

Equation for `m3` (`sbp` ~ `dbp` + `insurance`)

```
extract_eq(m3_train, use_coefs = TRUE,
           wrap = TRUE, terms_per_line = 2)
```

$$
\begin{aligned}
\widehat{\text{sbp}} &= 77.58 + 0.72(\text{dbp}) + \\
&\quad 1.11(\text{insurance}_{\text{Commercial}}) + 2.73(\text{insurance}_{\text{Medicare}}) + \\
&\quad 1.16(\text{insurance}_{\text{Uninsured}})
\end{aligned}
$$

- Predicted `sbp` by `m3` for a Commercial subject?
- `sbp` = 77.58 + 0.72*`dbp` + 1.11(1) + 2.73(0) + 1.16(0)
- `sbp` = 78.69 + 0.72*`dbp`
- For a Medicaid subject, `m3` predicts `sbp` = 77.58 + 0.72 `dbp`
- For a Medicare subject, `m3` predicts `sbp` = 80.31 + 0.72 `dbp`

Equation for m3 (sbp ~ dbp + insurance)

```
extract_eq(m3_train, use_coefs = TRUE,
           wrap = TRUE, terms_per_line = 2)
```

$$
\begin{aligned}
\widehat{\text{sbp}} &= 77.58 + 0.72(\text{dbp}) +\\
&\quad 1.11(\text{insurance}_{\text{Commercial}}) + 2.73(\text{insurance}_{\text{Medicare}}) +\\
&\quad 1.16(\text{insurance}_{\text{Uninsured}})
\end{aligned}
$$

- Predicted `sbp` by `m3` for a Commercial subject?
- `sbp` = 77.58 + 0.72*`dbp` + 1.11(1) + 2.73(0) + 1.16(0)
- `sbp` = 78.69 + 0.72*`dbp`
- For a Medicaid subject, `m3` predicts `sbp` = 77.58 + 0.72 `dbp`
- For a Medicare subject, `m3` predicts `sbp` = 80.31 + 0.72 `dbp`
- For an uninsured subject, `m3` predicts `sbp` = 78.74 + 0.72 `dbp`

Equation for m3 (sbp ~ dbp + insurance)

```
extract_eq(m3_train, use_coefs = TRUE,
           wrap = TRUE, terms_per_line = 2)
```

$$
\begin{aligned}
\widehat{\text{sbp}} &= 77.58 + 0.72(\text{dbp}) + \\
&\quad 1.11(\text{insurance}_{\text{Commercial}}) + 2.73(\text{insurance}_{\text{Medicare}}) + \\
&\quad 1.16(\text{insurance}_{\text{Uninsured}})
\end{aligned}
$$

- Predicted sbp by m3 for a Commercial subject?
- sbp = 77.58 + 0.72*dbp + 1.11(1) + 2.73(0) + 1.16(0)
- sbp = 78.69 + 0.72*dbp
- For a Medicaid subject, m3 predicts sbp = 77.58 + 0.72 dbp
- For a Medicare subject, m3 predicts sbp = 80.31 + 0.72 dbp
- For an uninsured subject, m3 predicts sbp = 78.74 + 0.72 dbp
- Note: only the intercept term varies by insurance in m3.

Equation for m4 (sbp ~ dbp * insurance)

```
extract_eq(m4_train, use_coefs = TRUE,
           wrap = TRUE, terms_per_line = 2)
```

$$
\begin{aligned}
\widehat{\text{sbp}} &= 60.26 + 0.94(\text{dbp})\ + \\
&\quad 23.54(\text{insurance}_{\text{Commercial}}) + 31.04(\text{insurance}_{\text{Medicare}})\ + \\
&\quad 25.78(\text{insurance}_{\text{Uninsured}}) - 0.29(\text{dbp} \times \text{insurance}_{\text{Commercial}})\ - \\
&\quad 0.38(\text{dbp} \times \text{insurance}_{\text{Medicare}}) - 0.32(\text{dbp} \times \text{insurance}_{\text{Uninsured}})
\end{aligned}
$$

- `m4` predicts, for a Commercial subject…

Equation for m4 (sbp ~ dbp * insurance)

```
extract_eq(m4_train, use_coefs = TRUE,
           wrap = TRUE, terms_per_line = 2)
```

$$
\begin{aligned}
\widehat{\operatorname{sbp}} &= 60.26 + 0.94(\operatorname{dbp})\ + \\
&\quad 23.54(\operatorname{insurance}_{\operatorname{Commercial}}) + 31.04(\operatorname{insurance}_{\operatorname{Medicare}})\ + \\
&\quad 25.78(\operatorname{insurance}_{\operatorname{Uninsured}}) - 0.29(\operatorname{dbp} \times \operatorname{insurance}_{\operatorname{Commercial}})\ - \\
&\quad 0.38(\operatorname{dbp} \times \operatorname{insurance}_{\operatorname{Medicare}}) - 0.32(\operatorname{dbp} \times \operatorname{insurance}_{\operatorname{Uninsured}})
\end{aligned}
$$

- `m4` predicts, for a Commercial subject…
- `sbp` = 60.26 + 0.94 * `dbp` + 23.54 (1) + 31.04 (0) + 25.78 (0) - 0.29 (dbp * 1) - 0.38 (dbp * 0) - 0.32 (dbp * 0)

Equation for m4 (sbp ~ dbp * insurance)

```
extract_eq(m4_train, use_coefs = TRUE,
           wrap = TRUE, terms_per_line = 2)
```

$$
\begin{aligned}
\widehat{\text{sbp}} &= 60.26 + 0.94(\text{dbp}) + \\
&\quad 23.54(\text{insurance}_{\text{Commercial}}) + 31.04(\text{insurance}_{\text{Medicare}}) + \\
&\quad 25.78(\text{insurance}_{\text{Uninsured}}) - 0.29(\text{dbp} \times \text{insurance}_{\text{Commercial}}) - \\
&\quad 0.38(\text{dbp} \times \text{insurance}_{\text{Medicare}}) - 0.32(\text{dbp} \times \text{insurance}_{\text{Uninsured}})
\end{aligned}
$$

- `m4` predicts, for a Commercial subject...
- `sbp` = 60.26 + 0.94 * `dbp` + 23.54 (1) + 31.04 (0) + 25.78 (0) - 0.29 (`dbp` * 1) - 0.38 (`dbp` * 0) - 0.32 (`dbp` * 0)
- `sbp` = (60.26 + 23.54) + (0.94 - 0.29) * `dbp`

Equation for m4 (sbp ~ dbp * insurance)

```
extract_eq(m4_train, use_coefs = TRUE,
           wrap = TRUE, terms_per_line = 2)
```

$$
\begin{aligned}
\widehat{\text{sbp}} &= 60.26 + 0.94(\text{dbp}) + \\
&\quad 23.54(\text{insurance}_{\text{Commercial}}) + 31.04(\text{insurance}_{\text{Medicare}}) + \\
&\quad 25.78(\text{insurance}_{\text{Uninsured}}) - 0.29(\text{dbp} \times \text{insurance}_{\text{Commercial}}) - \\
&\quad 0.38(\text{dbp} \times \text{insurance}_{\text{Medicare}}) - 0.32(\text{dbp} \times \text{insurance}_{\text{Uninsured}})
\end{aligned}
$$

- `m4` predicts, for a Commercial subject…
- `sbp` = 60.26 + 0.94 * `dbp` + 23.54 (1) + 31.04 (0) + 25.78 (0) - 0.29 (`dbp` * 1) - 0.38 (`dbp` * 0) - 0.32 (`dbp` * 0)
- `sbp` = (60.26 + 23.54) + (0.94 - 0.29) * `dbp`
- `sbp` = 83.80 - 0.65 `dbp` for Commercial subjects

Equation for m4 (sbp ~ dbp * insurance)

```
extract_eq(m4_train, use_coefs = TRUE,
           wrap = TRUE, terms_per_line = 2)
```

$$
\begin{aligned}
\widehat{\text{sbp}} &= 60.26 + 0.94(\text{dbp}) + \\
&\quad 23.54(\text{insurance}_{\text{Commercial}}) + 31.04(\text{insurance}_{\text{Medicare}}) + \\
&\quad 25.78(\text{insurance}_{\text{Uninsured}}) - 0.29(\text{dbp} \times \text{insurance}_{\text{Commercial}}) - \\
&\quad 0.38(\text{dbp} \times \text{insurance}_{\text{Medicare}}) - 0.32(\text{dbp} \times \text{insurance}_{\text{Uninsured}})
\end{aligned}
$$

- For Medicaid subjects, `sbp` = 60.26 + 0.94 * `dbp`
- For Medicare subjects, `sbp` = 91.30 + 0.56 * `dbp`
- For the uninsured, `sbp` = 86.04 + 0.62 * `dbp`
- So both the slope and the intercept are changing in `m4`

Training Sample Fit Quality

Model m3 (no interaction)

```
glance(m3_train) %>%
  select(r.squared, adj.r.squared, sigma, AIC, BIC) %>%
  kable(digits = c(3, 3, 1, 1, 1))
```

r.squared	adj.r.squared	sigma	AIC	BIC
0.224	0.22	16.1	5851.1	5878.4

Model m4 (with dbp-insurance interaction)

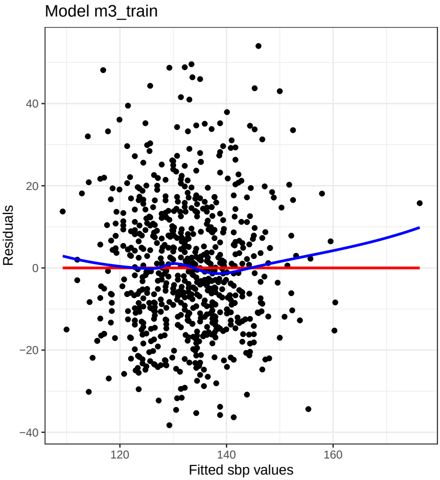
r.squared	adj.r.squared	sigma	AIC	BIC
0.236	0.229	16	5846	5886.9

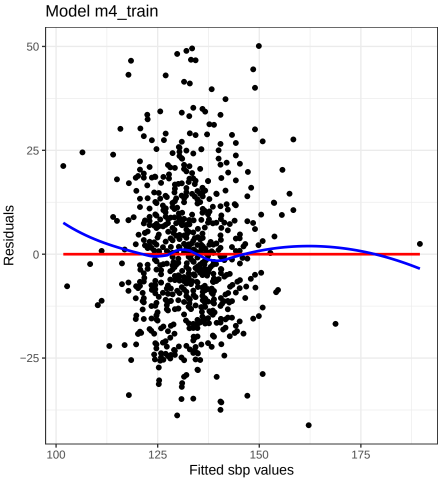
Augmenting and Testing Models m3 and m4

```
m3_train_aug <- augment(m3_train, data = dm_train)
m3_test_aug <- augment(m3_train, newdata = dm_test)

m4_train_aug <- augment(m4_train, data = dm_train)
m4_test_aug <- augment(m4_train, newdata = dm_test)
```

Residuals vs. Fitted Values Plots





Comparing performance on the training data

```
bind_rows(glance(m1_train), glance(m2_train),
          glance(m3_train), glance(m4_train)) %>%
  mutate(modname = c("m1", "m2", "m3", "m4")) %>%
  select(modname, r2 = r.squared, adj_r2 = adj.r.squared,
         sigma, AIC, BIC) %>%
  kable(digits = c(0, 3, 3, 2, 1, 1))
```

modname	r2	adj_r2	sigma	AIC	BIC
m1	0.220	0.219	16.12	5848.7	5862.3
m2	NA	NA	16.15	NA	NA
m3	0.224	0.220	16.11	5851.1	5878.4
m4	0.236	0.229	16.02	5846.0	5886.9

- The `glance()` function produces different results for a Bayesian `stan_glm()` model like `m2`, so we'll ignore that for now.

Comparing performance on the test data

Here are some fundamental summaries of absolute prediction error (APE) along with the root mean squared prediction error (RMSPE) for each of our models, in the **testing** sample.

Summary	Mean APE	Max APE	RMSPE
`m1_train`: `lm`	12.14	71.07	15.83
`m2_train`: `stan_glm`	12.14	71.06	15.83
`m3_train`: `dbp+insurance`	12.04	72.37	15.78
`m4_train`: `dbp*insurance`	11.95	71.37	15.65

- Which of these models displays the strongest predictive performance in our test sample?

Visualizing Categorical Data in `dm1000`

8 Categorical Variables from `dm1000`

```
dm_cat <- dm1000 %>%
  select(subject, sex, residence, insurance,
         tobacco, race_ethnicity, statin, eye_exam)
```

Codebook

- `subject` = ID value (treat as character)
- **sex** = Female or Male (no missing data)
- **insurance** = Medicare, Commercial, Medicaid, Uninsured
- **eye_exam** = 1 for eye examination in past year, else 0
- **statin** = 1 statin prescription in past year, else 0
- **race_ethnicity** = 4 levels (Hispanic or Latinx, Non-Hispanic White, Non-Hispanic Black, Non-Hispanic Asian)
- **residence** = 2 levels (Suburbs, Cleveland), some NA
- **tobacco** = 3 levels (Current, Former, Never), some NA

Using summary()

```
summary(dm_cat)
```

```
> summary(dm_cat)
   subject                 insurance      tobacco          statin
 Length:1000         Medicaid   :330   Current:274   Min.   :0.000
 Class :character    Commercial:196    Never  :343   1st Qu.:1.000
 Mode  :character    Medicare   :432   Former :367   Median :1.000
                     Uninsured : 42    NA's   : 16   Mean   :0.758
                                                     3rd Qu.:1.000
                                                     Max.   :1.000
    eye_exam          sex                      race_ethnicity     residence
 Min.   :0.000   Female:550   Non-Hispanic Black:533    Suburbs  :371
 1st Qu.:0.000   Male  :450   Hispanic or Latinx: 91    Cleveland:601
 Median :1.000                Non-Hispanic White:356    NA's     : 28
 Mean   :0.562                Non-Hispanic Asian: 20
 3rd Qu.:1.000
 Max.   :1.000
```

Using tabyl to tabulate a categorical variable

```
dm_cat %>% tabyl(tobacco) %>%
  adorn_pct_formatting() %>%
  adorn_totals()
```

```
 tobacco    n percent valid_percent
 Current  274   27.4%         27.8%
   Never  343   34.3%         34.9%
  Former  367   36.7%         37.3%
    <NA>   16    1.6%             -
   Total 1000       -             -
```

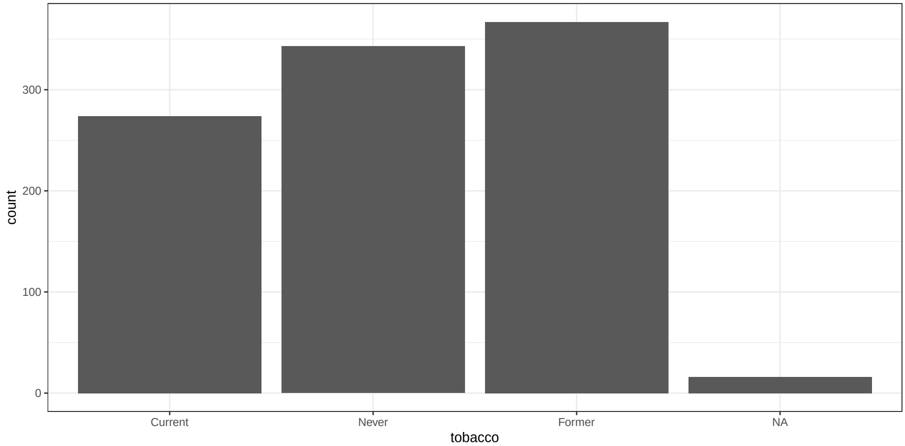
Using `count` to create a tibble of counts

```
dm_cat %>% count(tobacco)
```

```
# A tibble: 4 x 2
  tobacco     n
  <fct>   <int>
1 Current   274
2 Never     343
3 Former    367
4 <NA>       16
```

Using geom_bar to show a distribution

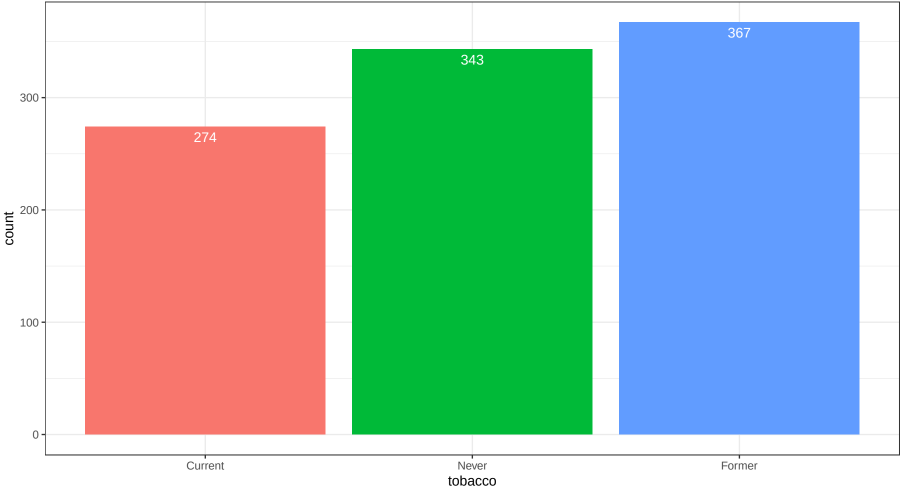
```
ggplot(dm_cat, aes(x = tobacco)) +
  geom_bar()
```



Augmenting the geom_bar result (code)

```
dm_cat %>% filter(complete.cases(tobacco)) %>%
  ggplot(data = ., aes(x = tobacco, fill = tobacco)) +
  geom_bar() +
  geom_text(aes(label = ..count..), stat = "count",
            vjust = 1.5, col = "white") +
  guides(fill = "none")
```

Augmenting the geom_bar result



Using tabyl to cross-tabulate two variables

```
dm_cat %>% tabyl(insurance, residence) %>%
  adorn_totals(where = c("row", "col"))
```

```
  insurance Suburbs Cleveland NA_ Total
   Medicaid     114       201  15   330
 Commercial      75       119   2   196
   Medicare     163       259  10   432
  Uninsured      19        22   1    42
      Total     371       601  28  1000
```

Using count to create a tibble of counts

```
dm_cat %>% count(statin, residence)
```

```
# A tibble: 6 x 3
  statin residence     n
   <dbl> <fct>     <int>
1      0 Suburbs      77
2      0 Cleveland   157
3      0 <NA>          8
4      1 Suburbs     294
5      1 Cleveland   444
6      1 <NA>         20
```

Were suburban residents more likely to have a statin prescription?

```
dm_cat %>%
  filter(complete.cases(statin, residence)) %>%
  tabyl(residence, statin)
```

```
 residence   0   1
   Suburbs  77 294
 Cleveland 157 444
```

Revise the order of the statin levels, add percentages

```
dm_cat %>% filter(complete.cases(statin, residence)) %>%
  mutate(statin = fct_relevel(factor(statin), "1", "0")) %>%
  tabyl(residence, statin)
```

```
 residence   1   0
   Suburbs 294  77
 Cleveland 444 157
```

```
dm_cat %>% filter(complete.cases(statin, residence)) %>%
  mutate(statin = fct_relevel(factor(statin), "1", "0")) %>%
  tabyl(residence, statin) %>%
  adorn_percentages(denom = "row") %>%
  adorn_pct_formatting()
```

```
 residence     1     0
   Suburbs 79.2% 20.8%
 Cleveland 73.9% 26.1%
```

Create using table instead

```
tab1 <- dm_cat %>%
  filter(complete.cases(statin, residence)) %>%
  mutate(statin = fct_relevel(factor(statin), "1", "0")) %$%
  table(residence, statin)
```

Assess 2x2 table (results on next slide)

```
Epi::twoby2(tab1)
```

twoby2 results

```
> Epi::twoby2(tab1)
2 by 2 table analysis:
------------------------------------------------------
Outcome   : 1
Comparing : Suburbs vs. Cleveland

           1   0    P(1) 95% conf. interval
Suburbs  294  77  0.7925    0.7482   0.8307
Cleveland 444 157  0.7388    0.7022   0.7723

                                   95% conf. interval
             Relative Risk: 1.0727    0.9996   1.1510
         Sample Odds Ratio: 1.3501    0.9903   1.8407
Conditional MLE Odds Ratio: 1.3497    0.9805   1.8679
    Probability difference: 0.0537   -0.0018   0.1065

             Exact P-value: 0.0638
        Asymptotic P-value: 0.0577
------------------------------------------------------
```

A three-by-four two-way table

```
dm_cat %>% filter(complete.cases(tobacco, insurance)) %>%
  tabyl(tobacco, insurance) %>%
  adorn_totals(where = c("row", "col"))
```

```
tobacco Medicaid Commercial Medicare Uninsured Total
Current      118         44       99        13   274
  Never      105         80      140        18   343
 Former      103         70      183        11   367
  Total      326        194      422        42   984
```

- 3 rows, 4 columns: hence, this is a 3 x 4 table
- It's a two-way table, because we are studying the association of two variables (`tobacco` and `insurance`)
- Can we compare the insurance percentages by tobacco group?

Compare insurance rates by tobacco group

```
dm_cat %>% filter(complete.cases(tobacco, insurance)) %>%
  tabyl(tobacco, insurance) %>%
  adorn_percentages(denominator = "row") %>%
  adorn_totals(where = "col") %>% kable(digits = 3)
```

tobacco	Medicaid	Commercial	Medicare	Uninsured	Total
Current	0.431	0.161	0.361	0.047	1
Never	0.306	0.233	0.408	0.052	1
Former	0.281	0.191	0.499	0.030	1

- Note that these are actually **proportions** and not percentages.
- Proportions fall between 0 and 1: multiply by 100 for percentages.

Insurance rates by tobacco group?

```
tab2 <- dm_cat %>%
  filter(complete.cases(tobacco, insurance)) %$%
  table(tobacco, insurance)

tab2
```

```
         insurance
tobacco   Medicaid Commercial Medicare Uninsured
  Current      118         44       99        13
  Never        105         80      140        18
  Former       103         70      183        11
```

```
chisq.test(tab2)
```

```
	Pearson's Chi-squared test

data:  tab2
X-squared = 25.592, df = 6, p-value = 0.0002651
```

Using count for three variables

```
dm_cat %>% count(sex, statin, eye_exam)
```

```
# A tibble: 8 x 4
  sex    statin eye_exam     n
  <fct>   <dbl>    <dbl> <int>
1 Female      0        0    68
2 Female      0        1    65
3 Female      1        0   176
4 Female      1        1   241
5 Male        0        0    61
6 Male        0        1    48
7 Male        1        0   133
8 Male        1        1   208
```

A three-way table

```
dm_cat %>% tabyl(statin, residence, sex) %>%
  adorn_title()
```

```
$Female
        residence
 statin   Suburbs Cleveland NA_
      0        42        87   4
      1       160       245  12

$Male
        residence
 statin   Suburbs Cleveland NA_
      0        35        70   4
      1       134       199   8
```

Flattening a three-way table

```
dm_cat %$%
  ftable(sex, residence, statin)
```

```
                 statin  0   1
sex    residence
Female Suburbs          42 160
       Cleveland        87 245
Male   Suburbs          35 134
       Cleveland        70 199
```

- Note that `ftable()` excludes the missing `residence` values by default.

Reminder of Today's Agenda

1. Developing Four Models for `sbp` using `dbp` (and `insurance`)
2. Fitting a Bayesian Linear Model with default priors (`m2`)
3. Including Insurance without (`m3`) and with (`m4`) interaction with `dbp` in linear models
4. Visualizing Categorical Data
5. Assessing Association in Cross-Tabulations