431 Class 11

Thomas E. Love, Ph.D.

2022-10-04

Today's Agenda

- Building models for `sbp_2` using `sbp_1` and `insur_1`
  - without an interaction term
  - with an interaction
- Comparing our four models (two from Class 10 and two today)

Today's R packages

```
library(broom)
library(equatiomatic)
library(haven)           ## import SPSS .sav file
library(rstanarm)        ## fit stan_glm() model
library(janitor)
library(kableExtra)
library(naniar)
library(patchwork)
library(tidyverse)

theme_set(theme_bw())
```

Today's Data

Today's data describe 1,500 adults with hypertension living in Cuyahoga County, whose (systolic) blood pressure was measured at baseline, and then again one year later. We also have information on (baseline) primary insurance, and other things.

- We created and partitioned the data back in Class 10

```
bp_full <- read_rds("c11/data/bp_full.Rds")
bp_train <- read_rds("c11/data/bp_train.Rds")
bp_test <- read_rds("c11/data/bp_test.Rds")
```

Research Questions

1. Can we build an effective model to predict `sbp_2` (SBP after a year) using `sbp_1` (SBP at baseline)? (addressed in class 10)
2. Is the effectiveness of such a model for prediction of `sbp_2` materially affected by whether we also include information about `ins_1` (Primary insurance at baseline)? (today)

Modeling Goals

Class 10

- Model `sbp_2` on the basis of `sbp_1`
  - using a linear regression model
  - using a (naive) Bayesian model

Today

- Model `sbp_2` using `sbp_1` and `ins_1`
  - without an interaction term
  - including an `sbp_1*ins_1` interaction term

Build models with **training** sample, evaluate performance in **testing** sample.

Previous models (`m1` and `m2`)

Fit in training sample, then evaluate in testing sample.

```r
m1_train <- lm(sbp_2 ~ sbp_1, data = bp_train)
m1_test_aug <- augment(m1_train, newdata = bp_test)

m2_train <- stan_glm(sbp_2 ~ sbp_1, data = bp_train, refresh = 0)
m2_test_aug <- bp_test |> select(record, sbp_2, sbp_1) |>
  mutate(.fitted = predict(m2_train, newdata = bp_test),
         .resid = sbp_2 - .fitted)
```

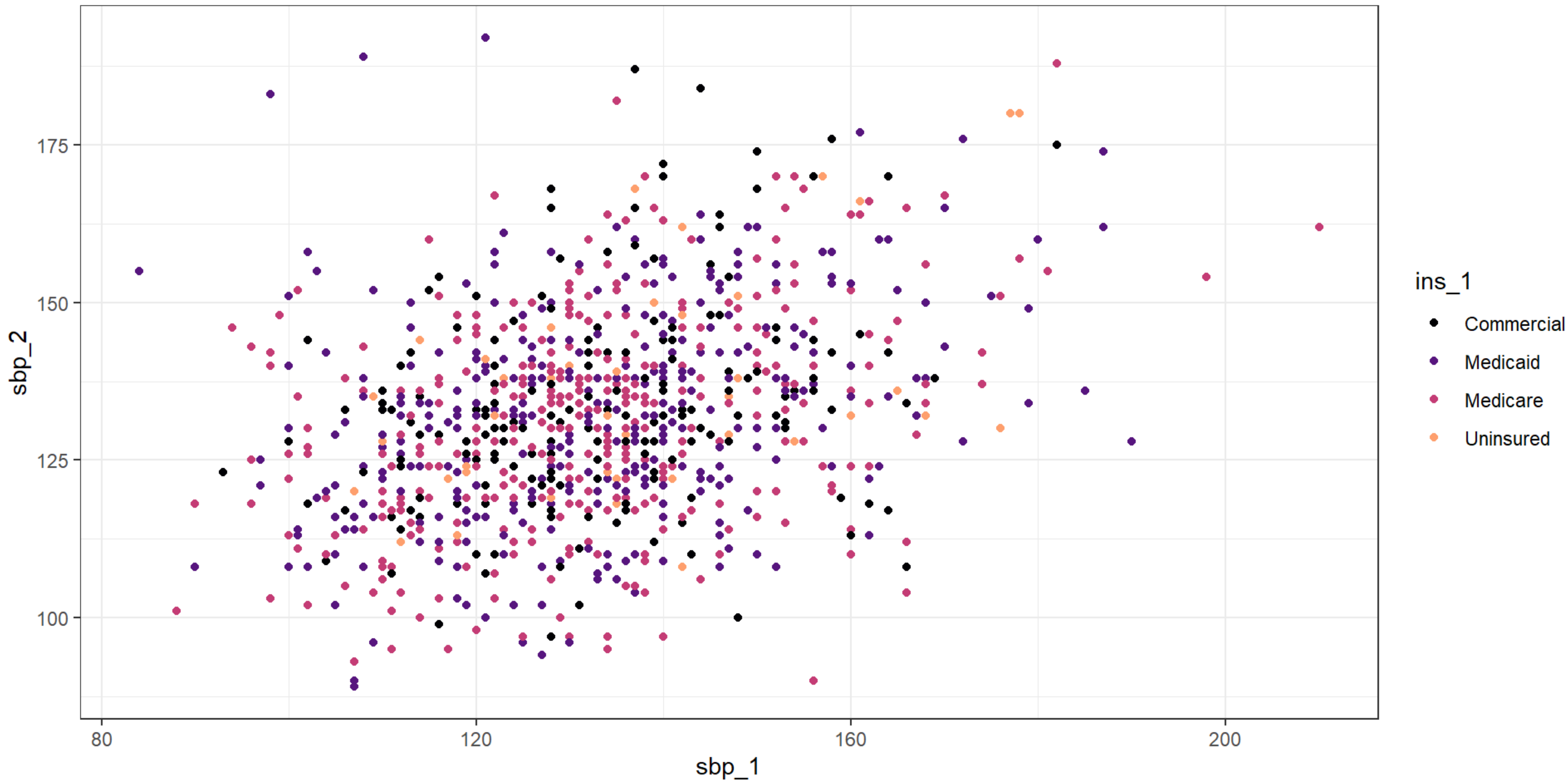
Which priors did we use in `m2_train`?

For more, visit https://mc-stan.org/rstanarm/articles/priors.html.

```r
prior_summary(m2_train)
```

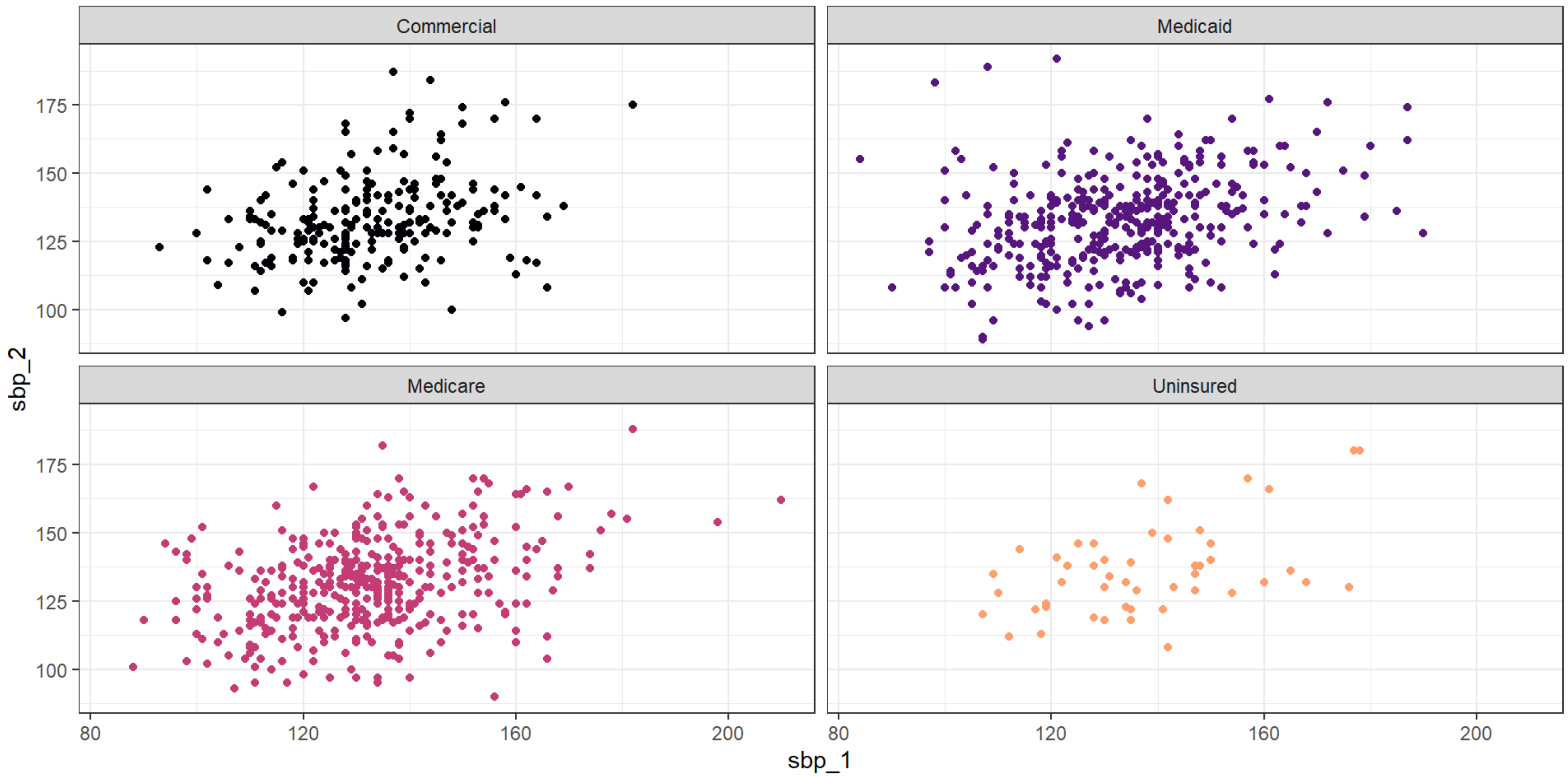
```
Priors for model 'm2_train'
------
Intercept (after predictors centered)
  Specified prior:
    ~ normal(location = 132, scale = 2.5)
  Adjusted prior:
    ~ normal(location = 132, scale = 41)

Coefficients
  Specified prior:
    ~ normal(location = 0, scale = 2.5)
  Adjusted prior:
    ~ normal(location = 0, scale = 2.4)

Auxiliary (sigma)
  Specified prior:
```

Add in `ins_1` information



Faceting by `ins_1` group

```
ggplot(data = bp_train, aes(x = sbp_1, y = sbp_2, col = ins_1)) +
  geom_point() + scale_color_viridis_d(option = "A", end = 0.8) +
  facet_wrap(~ ins_1) + guides(col = "none")
```



Two possible models

```
m3_train <- lm(sbp_2 ~ sbp_1 + ins_1, data = bp_train)
m4_train <- lm(sbp_2 ~ sbp_1 * ins_1, data = bp_train)
```

- What is the difference between `m3` and `m4`?
- Model `m3` does not include an interaction term, while `m4` does.
- How does this work in practice?

Equation for `m3`

```
m3_train <- lm(sbp_2 ~ sbp_1 + ins_1, data = bp_train)
```

```
extract_eq(m3_train, use_coefs = TRUE, operator_location = "start",
           wrap = TRUE, terms_per_line = 2, coef_digits = 2)
```

$$
\begin{aligned}
\widehat{\text{sbp\_2}} &= 89.36 + 0.33(\text{sbp\_1}) \\
&\quad - 0.83(\text{ins\_1}_{\text{Medicaid}}) - 2.41(\text{ins\_1}_{\text{Medicare}}) \\
&\quad + 1.38(\text{ins\_1}_{\text{Uninsured}})
\end{aligned}
$$

In model `m3`, the intercept term of the `sbp_1`-`sbp_2` relationship varies depending on insurance.

Model m3 by Insurance Type

$$\widehat{\text{sbp\_2}} = 89.36 + 0.33(\text{sbp\_1}) - 0.83(\text{ins\_1}_{\text{Medicaid}}) - 2.41(\text{ins\_1}_{\text{Medicare}}) + 1.38(\text{ins\_1}_{\text{Uninsured}})$$

Insurance	Estimated sbp_2
Commmercial	89.36 + 0.33 sbp_1
Medicaid	??
Medicare	??
Uninsured	??

Model m3 by Insurance Type

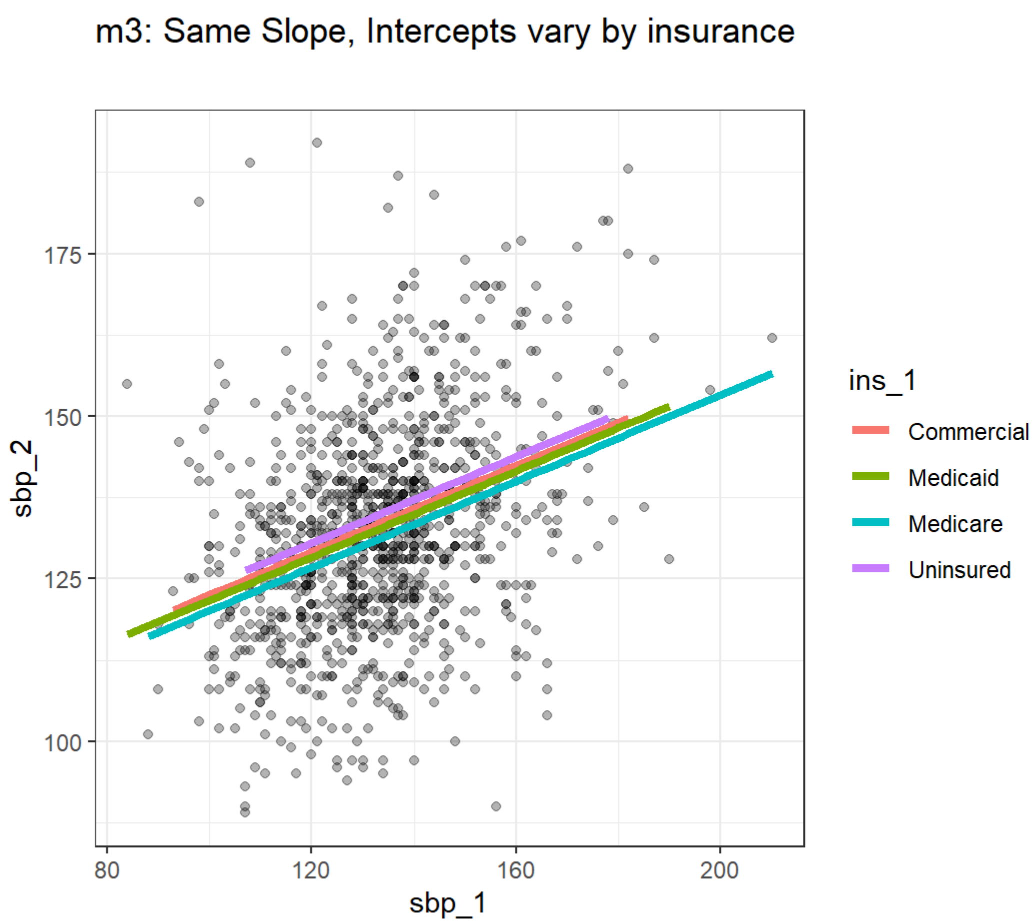
$$\widehat{\text{sbp\_2}} = 89.36 + 0.33(\text{sbp\_1}) - 0.83(\text{ins\_1}_{\text{Medicaid}}) - 2.41(\text{ins\_1}_{\text{Medicare}}) + 1.38(\text{ins\_1}_{\text{Uninsured}})$$

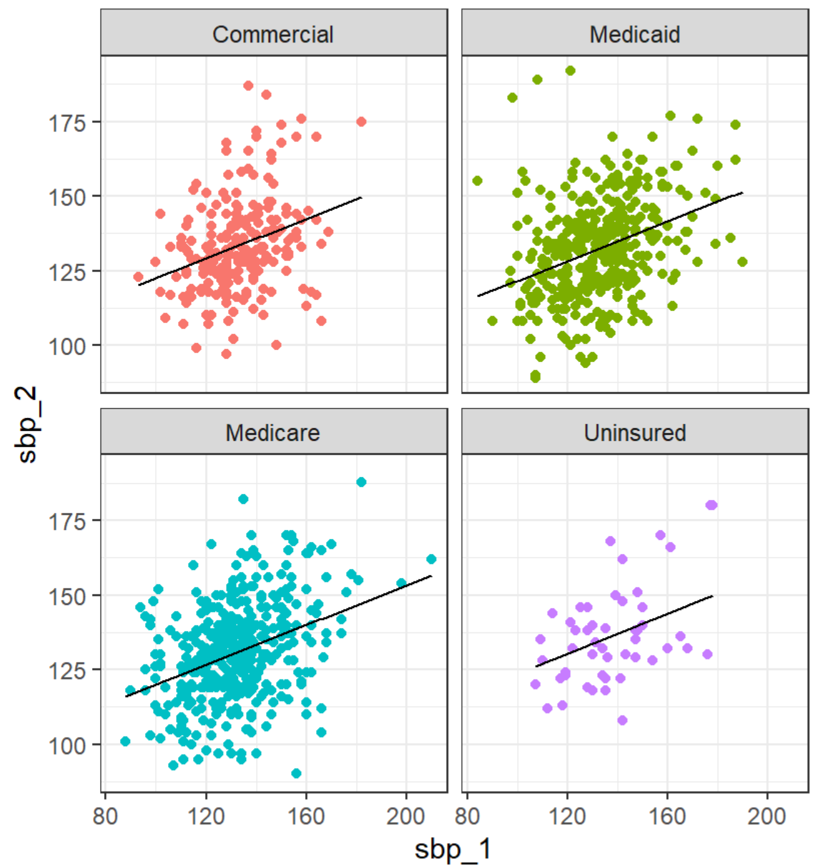
Insurance	Estimated sbp_2
Commmercial	89.36 + 0.33 sbp_1
Medicaid	(89.36 - 0.83) + 0.33 sbp_1 = **88.53** + 0.33 sbp_1
Medicare	(89.36 - 2.41) + 0.33 sbp_1 = **86.95** + 0.33 sbp_1
Uninsured	(89.36 + 1.38) + 0.33 sbp_1 = **90.74** + 0.33 sbp_1

The m3 model (pictured)

```
m3_train_aug <- augment(m3_train, data = bp_train)

p1 <- ggplot(m3_train_aug, aes(x = sbp_1, y = sbp_2, group = ins_1)) +
  geom_point(alpha = 0.3) +
  geom_line(aes(x = sbp_1, y = .fitted, col = ins_1), lwd = 1.5) +
  labs(title = "m3: Same Slope, Intercepts vary by insurance")

p2 <- ggplot(m3_train_aug, aes(x = sbp_1, y = sbp_2,
                            col = ins_1, group = ins_1)) +
  geom_point() + geom_line(aes(x = sbp_1, y = .fitted), col = "black") +
  facet_wrap( ~ ins_1) + guides(col = "none")

p1 + p2
```

The m3 model (pictured)





Tidied Model `m3` coefficients

Again, in model `m3`, only the intercept of the `sbp_1` to `sbp_2` model varies depending on the `ins_1` category.

```
tidy(m3_train, conf.int = TRUE, conf.level = 0.90) |>
  select(term, estimate, std.error, conf.low, conf.high) |>
  kbl(digits = c(0, 2, 2, 2, 2)) |> kable_styling(font_size = 28)
```

term	estimate	std.error	conf.low	conf.high
(Intercept)	89.36	3.82	83.07	95.64
sbp_1	0.33	0.03	0.29	0.38
ins_1Medicaid	-0.83	1.28	-2.95	1.28
ins_1Medicare	-2.41	1.28	-4.51	-0.30
ins_1Uninsured	1.38	2.41	-2.59	5.34

Equation for m4

```r
extract_eq(m4_train, use_coefs = TRUE, operator_location = "start", wrap = 
            terms_per_line = 1, coef_digits = 2, font_size = "small")
```

$$
\begin{aligned}
\widehat{\text{sbp\_2}} &= 90.33 \\
&\quad + 0.32(\text{sbp\_1}) \\
&\quad + 1.56(\text{ins\_1}_{\text{Medicaid}}) \\
&\quad - 4.86(\text{ins\_1}_{\text{Medicare}}) \\
&\quad - 16.75(\text{ins\_1}_{\text{Uninsured}}) \\
&\quad - 0.02(\text{sbp\_1} \times \text{ins\_1}_{\text{Medicaid}}) \\
&\quad + 0.02(\text{sbp\_1} \times \text{ins\_1}_{\text{Medicare}}) \\
&\quad + 0.13(\text{sbp\_1} \times \text{ins\_1}_{\text{Uninsured}})
\end{aligned}
$$

Model m4 by Insurance Type

$$\widehat{\text{sbp\_2}} = 90.33 + 0.32(\text{sbp\_1}) + 1.56(\text{ins\_1}_{\text{Medicaid}}) - 4.86(\text{ins\_1}_{\text{Medicare}}) - 16.75(\text{ins\_1}_{\text{Uninsured}}) - 0.02(\text{sbp\_1} \times \text{ins\_1}_{\text{Medicaid}}) + 0.02(\text{sbp\_1} \times \text{ins\_1}_{\text{Medicare}}) + 0.13(\text{sbp\_1} \times \text{ins\_1}_{\text{Uninsured}})$$

Insurance	Estimated sbp_2
Commmercial	90.33 + 0.32 sbp_1
Medicaid	??
Medicare	??
Uninsured	??

Model m4 by Insurance Type

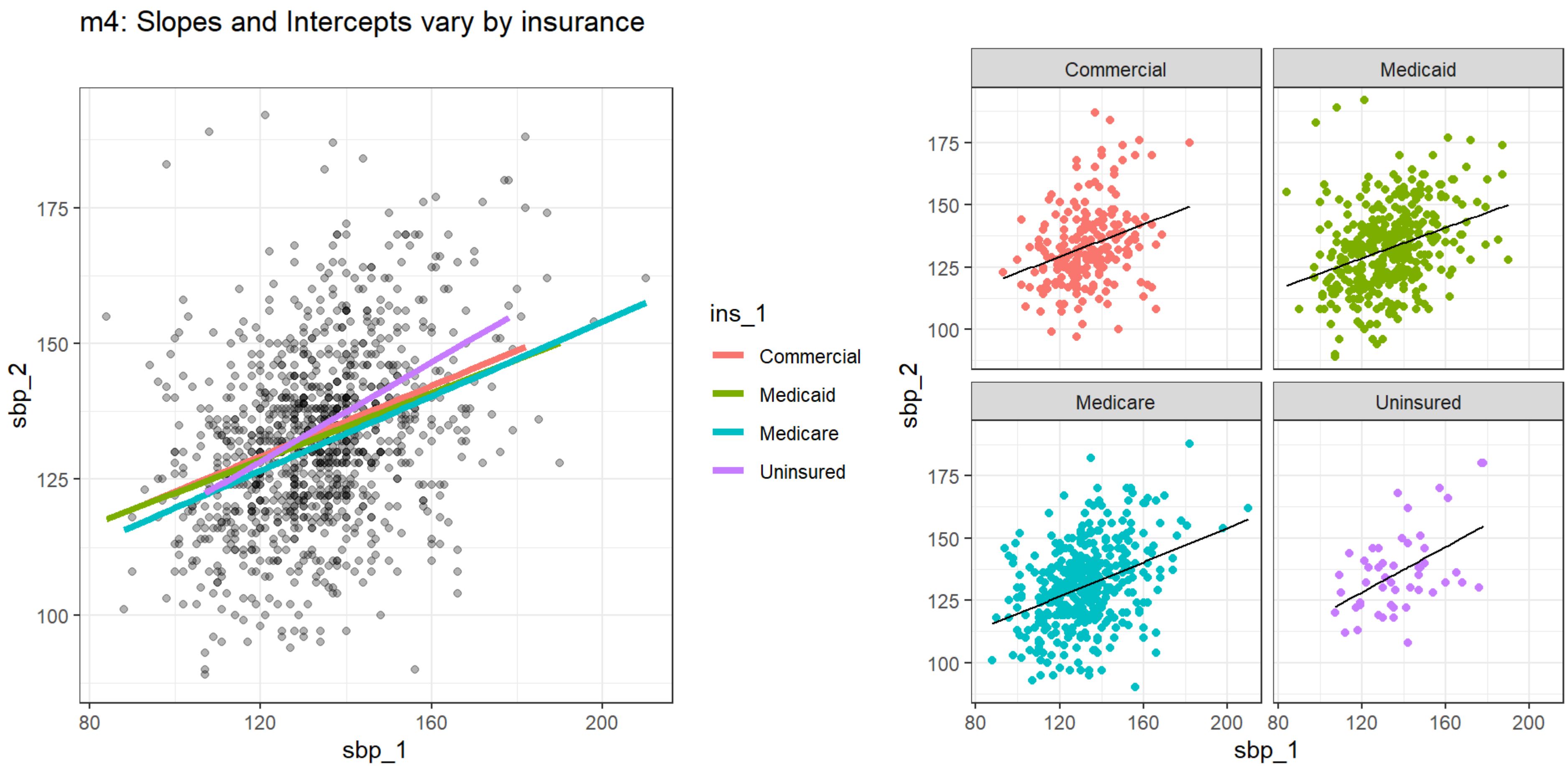
$$\widehat{\text{sbp\_2}} = 90.33 + 0.32(\text{sbp\_1}) + 1.56(\text{ins\_1}_{\text{Medicaid}}) \\ - 4.86(\text{ins\_1}_{\text{Medicare}}) - 16.75(\text{ins\_1}_{\text{Uninsured}}) - 0.02(\text{sbp\_1} \times \text{ins\_1}_{\text{Medicaid}}) \\ + 0.02(\text{sbp\_1} \times \text{ins\_1}_{\text{Medicare}}) + 0.13(\text{sbp\_1} \times \text{ins\_1}_{\text{Uninsured}})$$

Insurance	Estimated sbp_2
Commmercial	90.33 + 0.32 sbp_1
Medicaid	(90.33 + 1.56) + (0.32 - 0.02) sbp_1 = **91.89** + **0.30** sbp_1
Medicare	(90.33 - 4.86) + (0.32 + 0.02) sbp_1 = **85.47** + **0.34** sbp_1
Uninsured	(90.33 - 16.75) + (0.32 + 0.13) sbp_1 = **73.58** + **0.45** sbp_1

The m4 model (pictured)

```
m4_train_aug <- augment(m4_train, data = bp_train)

p1 <- ggplot(m4_train_aug, aes(x = sbp_1, y = sbp_2, group = ins_1)) +
  geom_point(alpha = 0.3) +
  geom_line(aes(x = sbp_1, y = .fitted, col = ins_1), lwd = 1.5) +
  labs(title = "m4: Slopes and Intercepts vary by insurance")

p2 <- ggplot(m4_train_aug, aes(x = sbp_1, y = sbp_2,
                               col = ins_1, group = ins_1)) +
  geom_point() + geom_line(aes(x = sbp_1, y = .fitted), col = "black") +
  facet_wrap( ~ ins_1) + guides(col = "none")

p1 + p2
```

The m4 model (pictured)



431 CASE WESTERN RESERVE UNIVERSITY

Models m3 and m4

```
m3_train <- lm(sbp_2 ~ sbp_1 + ins_1, data = bp_train)
m4_train <- lm(sbp_2 ~ sbp_1 * ins_1, data = bp_train)
```

- What is the difference between `m3` and `m4`?
  - Model `m3` will allow **only the intercept** term of the `sbp_1`-`sbp_2` relationship to vary depending on insurance.
  - Model `m4` will allow **both the slope and intercept** of the `sbp_1`-`sbp_2` relationship to vary depending on insurance.

Tidied Model m4 coefficients

```r
tidy(m4_train, conf.int = TRUE, conf.level = 0.90) |>
  select(term, estimate, std.error, conf.low, conf.high) |>
  kbl(digits = c(0, 2, 2, 2, 2)) |> kable_styling(font_size = 24)
```

term	estimate	std.error	conf.low	conf.high
(Intercept)	90.33	9.30	75.02	105.64
sbp_1	0.32	0.07	0.21	0.44
ins_1Medicaid	1.56	11.05	-16.63	19.74
ins_1Medicare	-4.86	10.96	-22.90	13.18
ins_1Uninsured	-16.75	19.35	-48.61	15.10
sbp_1:ins_1Medicaid	-0.02	0.08	-0.15	0.12
sbp_1:ins_1Medicare	0.02	0.08	-0.12	0.15
sbp_1:ins_1Uninsured	0.13	0.14	-0.10	0.36

Fit within the Training Sample

Model `m3` (no interaction)

```
glance(m3_train) |> select(r.squared, sigma, AIC, df, df.residual, nobs) |>
  kbl(digits = c(3, 1, 1, 0, 0, 0)) |> kable_styling(font_size = 32)
```

r.squared	sigma	AIC	df	df.residual	nobs
0.128	15.2	8707	4	1045	1050

Model `m4` (with `sbp_1`-`insurance` interaction)

```
glance(m4_train) |> select(r.squared, sigma, AIC, df, df.residual, nobs) |>
  kbl(digits = c(3, 1, 1, 0, 0, 0)) |> kable_styling(font_size = 32)
```

r.squared	sigma	AIC	df	df.residual	nobs
0.129	15.3	8711.5	7	1042	1050

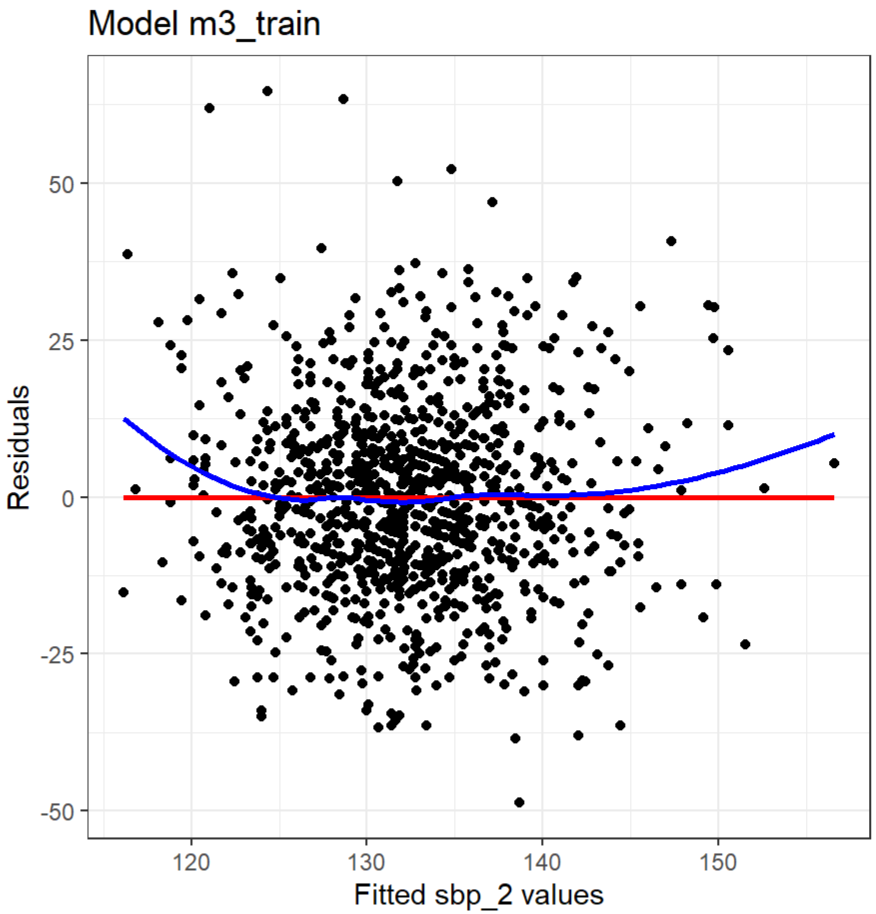
Augmenting and Testing Models m3 and m4

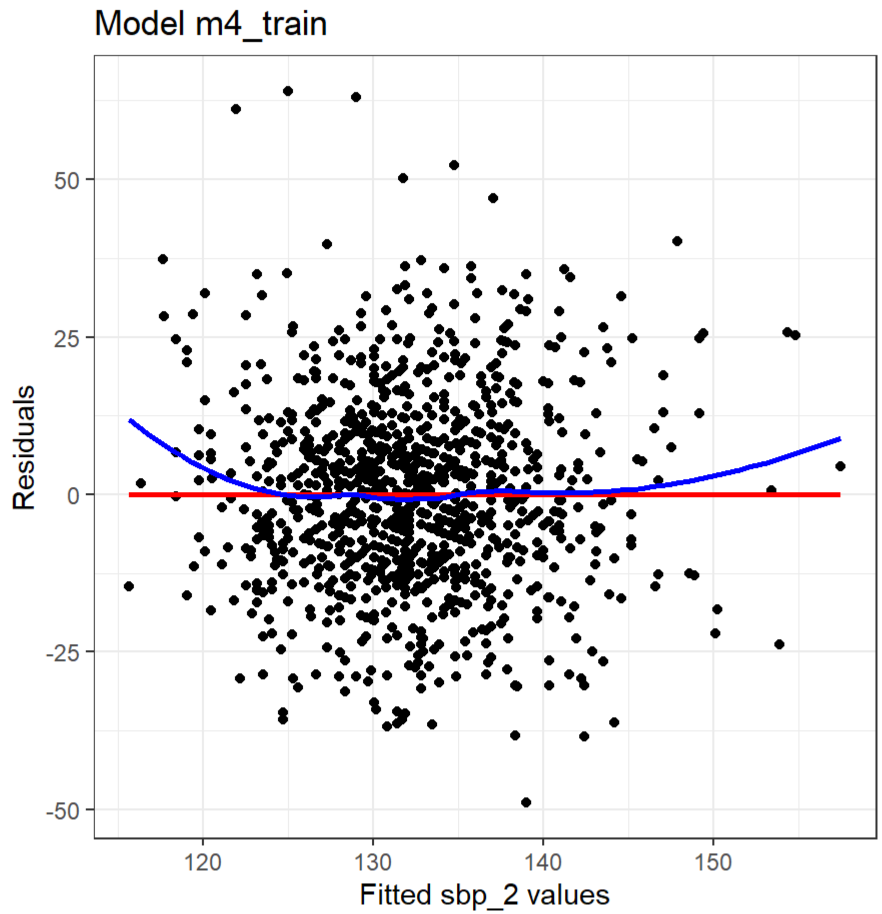
```
## in the training sample (for residual plots)

m3_train_aug <- augment(m3_train, data = bp_train)
m4_train_aug <- augment(m4_train, data = bp_train)

# in the test sample (calculating prediction errors)

m3_test_aug <- augment(m3_train, newdata = bp_test)
m4_test_aug <- augment(m4_train, newdata = bp_test)
```

Residuals vs. Fitted Values Plots

```
p1 <- ggplot(m3_train_aug, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_smooth(method = "lm", col = "red",
              formula = y ~ x, se = FALSE) +
  geom_smooth(method = "loess", col = "blue",
              formula = y ~ x, se = FALSE) +
  theme(aspect.ratio = 1) +
  labs(title = "Model m3_train",
       x = "Fitted sbp_2 values", y = "Residuals")

p2 <- ggplot(m4_train_aug, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_smooth(method = "lm", col = "red",
              formula = y ~ x, se = FALSE) +
  geom_smooth(method = "loess", col = "blue",
              formula = y ~ x, se = FALSE) +
  theme(aspect.ratio = 1) +
  labs(title = "Model m4_train",
       x = "Fitted sbp_2 values", y = "Residuals")
```

Residuals vs. Fitted Values Plots





m3 and m4: Same predictions?

```
t1 <- bind_cols(m3_train_aug$record, m3_train_aug$ins_1, m3_train_aug$.fitt
                m4_train_aug$.fitted)

names(t1) <- c("record", "ins_1", "m3_fit", "m4_fit")

p1 <- ggplot(data = t1, aes(x = m3_fit, y = m4_fit)) +
  geom_abline(aes(col = "black"), intercept = 0, slope = 1) +
  geom_point(size = 2) +
  theme(aspect.ratio = 1) +
  labs(title = "Figure 1. Predicted sbp_2 from m3, m4")

p2 <- ggplot(data = t1, aes(x = m3_fit, y = m4_fit, col = ins_1)) +
  geom_abline(aes(col = "black"), intercept = 0, slope = 1) +
  geom_point(size = 2) +
  theme(aspect.ratio = 1) +
  facet_wrap( ~ ins_1) +
  guides(col = "none") +
  labs(title = "Figure 2. Predicted sbp_2 by ins_1")
```

m3 and m4: Same predictions?

Figure 1. Predicted sbp_2 from m3, m4

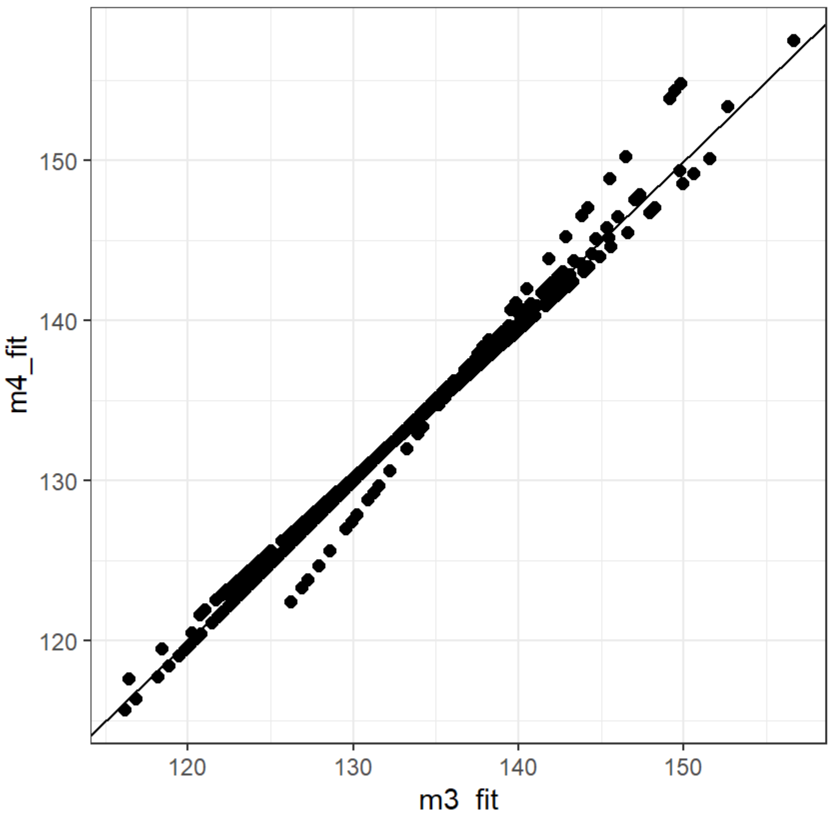
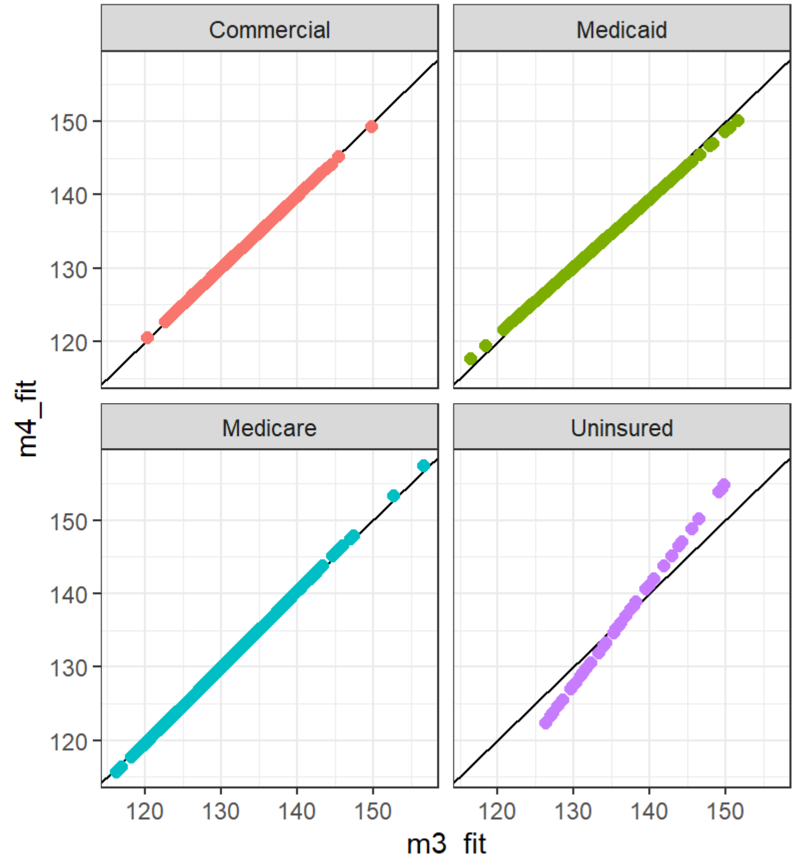


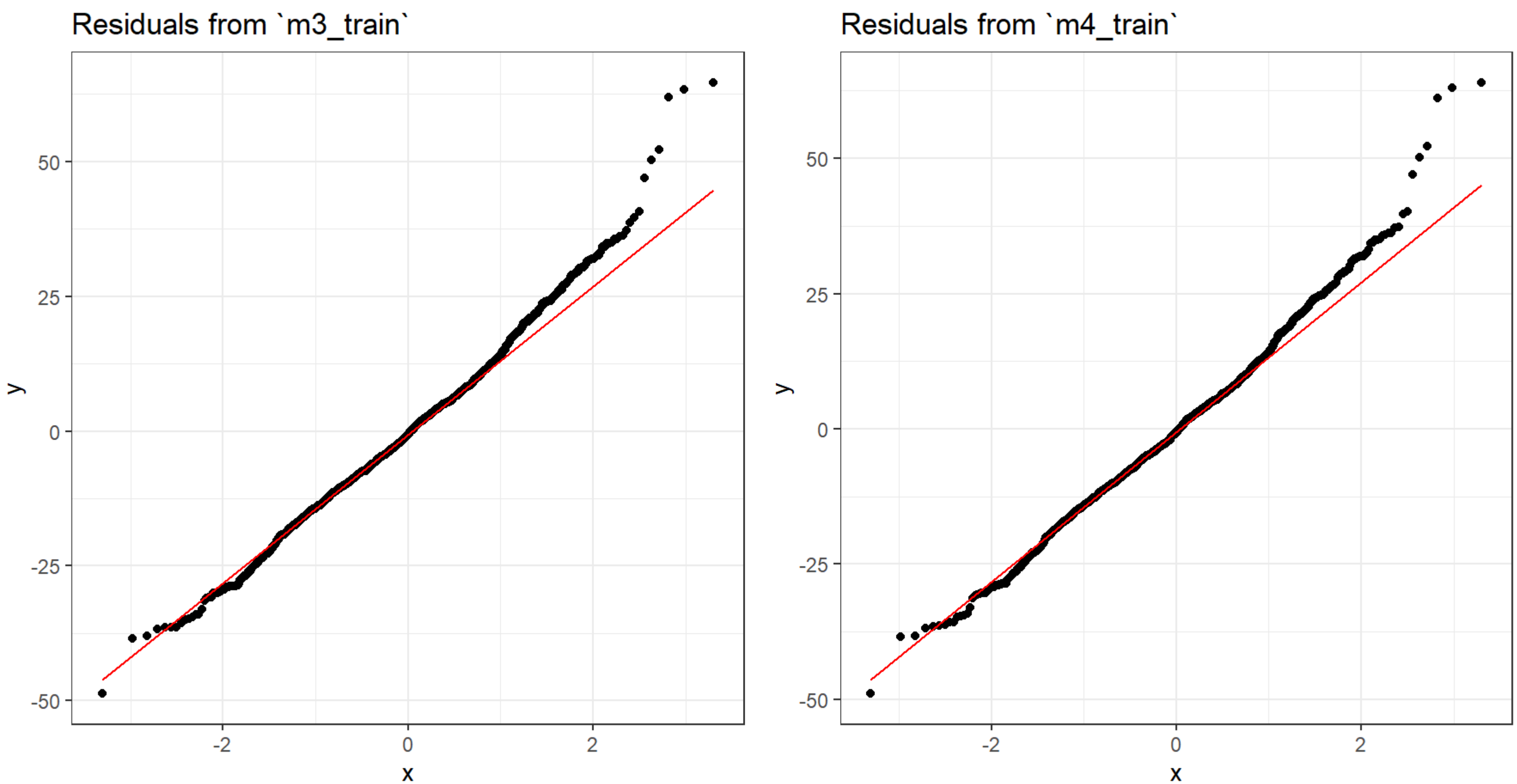
Figure 2. Predicted sbp_2 by ins_1



Normality of Residuals?

```r
p1 <- ggplot(m3_train, aes(sample = .resid)) +
  geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1) +
  labs(title = "Residuals from `m3_train`")

p2 <- ggplot(m4_train, aes(sample = .resid)) +
  geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1) +
  labs(title = "Residuals from `m4_train`")

p1 + p2
```

Normality of Residuals?



Training Set Performance

```r
bind_rows(glance(m1_train), broom.mixed::glance(m2_train), glance(m3_train)
          glance(m4_train)) |>
  mutate(model = c("m1", "m2", "m3", "m4")) |>
  select(model, r2 = r.squared, sigma, AIC) |>
  kbl(digits = c(0, 3, 2, 1)) |> kable_styling(font_size = 28)
```

model	r2	sigma	AIC
m1	0.123	15.26	8706.4
m2	NA	15.26	NA
m3	0.128	15.24	8707.0
m4	0.129	15.25	8711.5

- `glance()` produces different summaries for a Bayesian `stan_glm()` model like `m2`.

Test Sample Results for Model `m3`

```r
m3_test_aug <- augment(m3_train, newdata = bp_test)

## Summarize absolute prediction errors
mosaic::favstats(~ abs(.resid), data = m3_test_aug) |>
  kbl(digits = 2) |> kable_styling(font_size = 28)
```

min	Q1	median	Q3	max	mean	sd	n	missing
0.02	3.87	8.54	16.03	59.24	11.15	9.73	450	0

```r
## Summarize squared prediction errors
mosaic::favstats(~ .resid^2, data = m3_test_aug) |>
  kbl(digits = 2) |> kable_styling(font_size = 28)
```

min	Q1	median	Q3	max	mean	sd	n	missing
0	14.99	72.94	256.9	3509.12	218.71	392.28	450	0

- MAPE = 11.15, max APE = 59.24
- RMSPE = $\sqrt{218.71}$ = 14.79

Test Sample Results for Model m4

```r
m4_test_aug <- augment(m4_train, newdata = bp_test)

## Obtain mean, maximum absolute error and root mean squared error
m4_test_aug |> select(.resid) |>
  summarize(MAPE = mean(abs(.resid)), maxAPE = max(abs(.resid)),
            RMSPE = sqrt(mean(.resid^2))) |>
  kbl(digits = 2) |> kable_styling(font_size = 32)
```

MAPE	maxAPE	RMSPE
11.14	59.38	14.77

Test Sample Correlation(fitted, actual)

Pearson correlation between fitted predictions and actual `sbp_2` within the test sample.

- We could also square this to get an R^2 result.

```
round_half_up(cor(m1_test_aug$.fitted, m1_test_aug$sbp_2),4)
```

```
[1] 0.3875
```

```
round_half_up(cor(m2_test_aug$.fitted, m2_test_aug$sbp_2),4)
```

```
[1] 0.3875
```

```
round_half_up(cor(m3_test_aug$.fitted, m3_test_aug$sbp_2),4)
```

```
[1] 0.391
```

```
round_half_up(cor(m4_test_aug$.fitted, m4_test_aug$sbp_2),4)
```

```
[1] 0.3945
```

Comparing performance on the test data

- Which model performs best in our test sample?

Summary	MAPE	Max APE	RMSPE	Cor(Fit,Obs)
`m1`: `lm sbp_1`	11.17	58.02	14.81	0.3875
`m2`: `stan_glm`	11.17	58.02	14.81	0.3875
`m3`: `sbp_1+ins`	11.15	59.24	14.79	0.391
`m4`: `sbp_1*ins`	11.14	59.38	14.77	0.3945

Session Information

```
sessionInfo()
```

```
R version 4.2.1 (2022-06-23 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 22000)

Matrix products: default

locale:
[1] LC_COLLATE=English_United States.utf8
[2] LC_CTYPE=English_United States.utf8
[3] LC_MONETARY=English_United States.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.utf8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base
```