STA610 Case Study 1

Emily Gentles (Presenter)   Weiyi Liu (Writer)   Jack McCarthy (Programmer)
Qinzhe Wang (Coordinator & Checker)

15 October, 2021

Introduction

Prescription opioid abuse has recently become an epidemic in the United States. The price of illicit prescription opioids indicates the supply-demand relationship of the drug. This case study aims to explore the relationship between the unit price of drugs and other factors such as the quantity purchased, the location of the transaction, and strength of the drug. More specifically, our group's interest is to explore the factors related to the cost per milligram and the heterogeneity in the region. The dataset we will be using is provided by StreetRx, a reporting tool for people at large to anonymously report the price they paid or heard for diverted prescription drugs.

Our drug of interest is Morphine which is used to "relieve moderate to severe pain and may be habit-forming," especially with prolonged use (MedlinePlus).

EDA

Missing Values

The subset of the StreetRx dataset pertaining to Morphine contains 9,268 observations with 13 variables. There are 13,443 empty cells, including both missing values and blank entries. To maintain the statistical power and avoid bias, our group decided to recode both the empty cells and "0 Reporter did not answer this question" in `Primary_Reason` (5061 in total) as "8 Prefer not to answer" and recode the empty cells in `source` (3942 in total) as "Blank" because of the high missing rates. Then, we removed other rows with missing values.

Additionally, we removed non-positive price values as well as price values greater than 10. Since the data is self-reported, these extremely expensive prices are likely due to users misunderstanding the system and reporting total price instead of unit price. The number of observations is now 5,582.

Response Variable: Price per milligram

Distribution of morphine ppm

Distribution of morphine log(ppm)
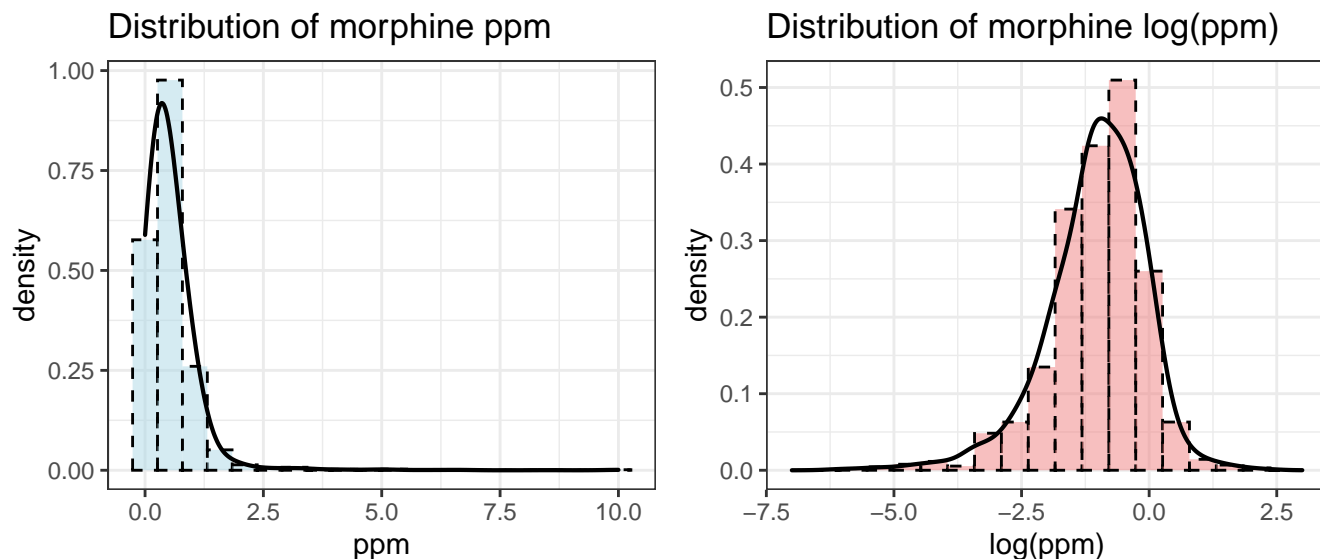
Whether we fit a hierarchical model or linear regression, the response variable should be normally distributed. Although the normality assumption pertains to the conditional distribution of our response variable, it's still beneficial to check the assumption for the marginal distribution as a very skewed marginal distribution could persist and affect the model's resulting conditional distribution. From the histogram on the left, the distribution of `ppm` is clearly right-skewed. Since `ppm` is strictly non-negative, a log transformation may be appropriate. We can see that the distribution of `log(ppm)`, given above, appears to be much closer to the desired normal distribution.

**Grouping Variable: city, state, and USA_region**

Since we want to analyze the heterogeneity in pricing by location, we have three choices of grouping variables, `city`, `state`, and `USA_region`.

City

There are 1642 unique `city` values, and many cities have small sample size (i.e. less than 5 observations). We decide not to use `city` as the grouping variable (see appendix).

State

As for the state, we examined the sample sizes in each group and decided to out filter Puerto Rico and Vermont because they have less than 5 observations.
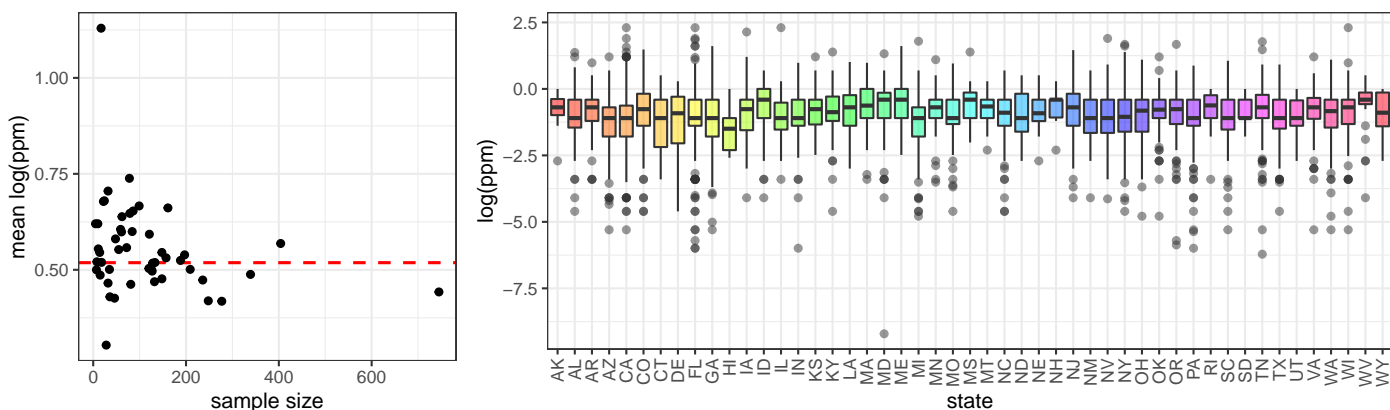
We then inspect the state-level differences more closely by plotting the group-level means against the sample sizes. We observed that the within-state means for states with smaller sample sizes vary a lot, while the within-state means for states with higher sample sizes in general adhere more closely to the grand mean. This is conducive to the borrowing of information between states with a hierarchical model. From the above boxplot of `log(ppm)` against `state`, it is also evident that the `log(ppm)` distributions differ across states. This indicates the potential state-level differences in drug prices. Therefore, we decide to use state as our grouping variable at this stage.

Region

From the boxplot we see that the `log(ppm)` distributions differ slightly across regions, though not as much as across states. We may also consider using region as the grouping variable.

|	usa_region	n	mean
1	Midwest	1168	−1.056
2	Northeast	674	−0.962
3	South	1953	−0.972
4	West	1773	−1.151
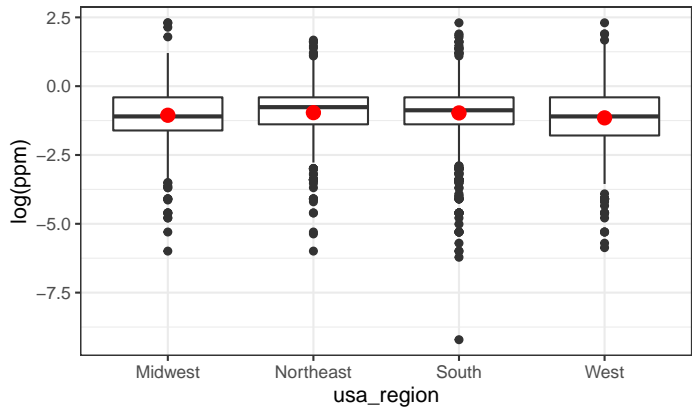
Date (price_date)

As for the `price_date`, we noticed some observations are prior to the establishment of StreetRx, which are likely incorrect inputs. We dropped the observations before 2010. For the remaining observations, we came up with two ways of data cleaning. The first choice is to choose a starting date and convert the feature as the date differences (`date_diff`) from that starting date. The second choice is to split this date variable into two components, `year` and `quarter`, to explore the trend of unit drug price over time and the seasonality.

Our visualizations suggested there is no clear indication that the log value of per milligram price of morphine varies along with `date_diff`. However, for different `year` and `quarter`, the `log(ppm)` value varies slightly (see appendix).

Bulk_purchase & Source
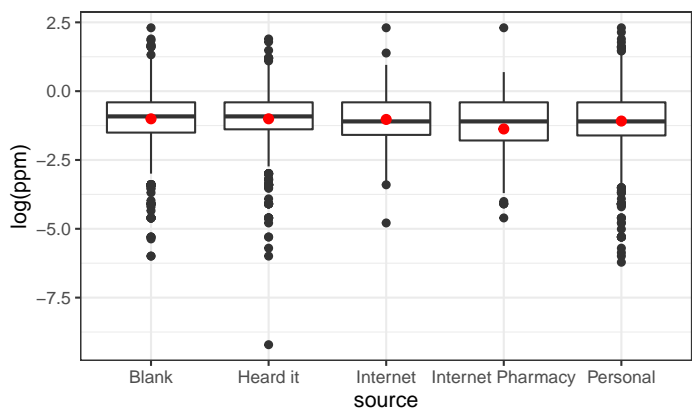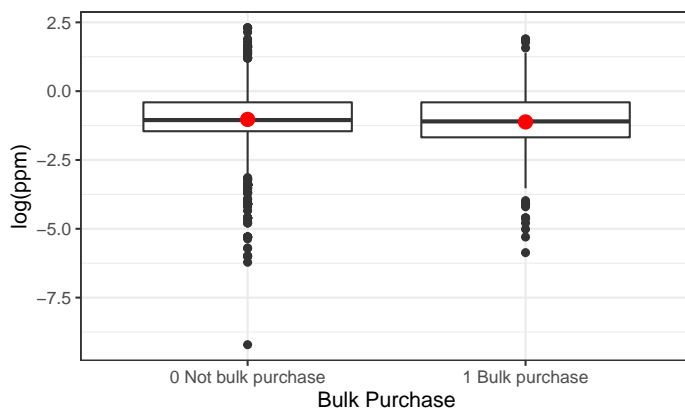
There is no need to conduct any data cleaning on `bulk_purchase`. And from the boxplot (see appendix), there is a slight trend that the drug price may be lower if purchased in bulk. Therefore, `bulk_purchase` might be a potential predictor.

For the feature `source`, we have recoded the missing value as "Blank" and the name of websites as "Internet". We also dropped the only observation whose `source` is "Drug Forum". The boxplot shows that the `log(ppm)` value varies among different sources (see appendix).

Dosage Strength & Primary Reason
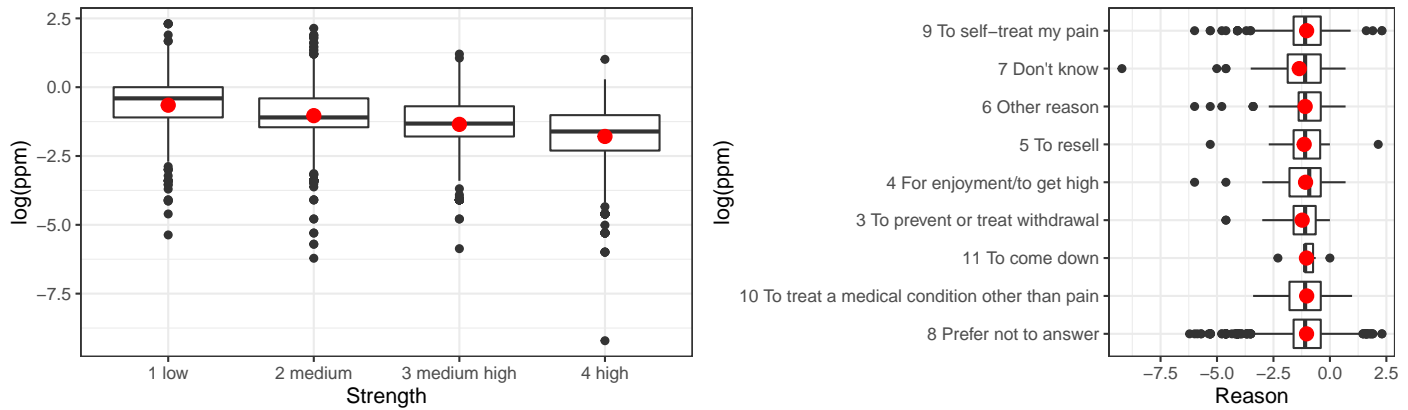
From the scatter plot of `log(ppm)` against `mgstr`, there is a slight trend that the larger the dosage strength, the smaller the per milligram price. We have also noticed that `mgstr` only takes 16 discrete values. Therefore, we decided to transform it into 4 levels ("low", "medium", "medium high", and "high") based on the 0.25, 0.5, and 0.75 quantiles of `mgstr`. From the boxplot, the trend that the `log(ppm)` values decrease as the dosage strength increases is more clear when using these new levels.

For `primary_reason`, we have converted the empty cells and "0 Reporter did not answer this question" to "8 Prefer not to answer". The `log(ppm)` value varies among different reasons for purchasing morphine (see appendix).

Model

Initial Model & Model Selection

The goal of our analysis is to investigate factors related to the per milligram price of morphine and explore heterogeneity in pricing by location. As discussed in the EDA part, we do not have enough data to estimate the effects at the city level. Meanwhile, the drug prices do not seem to change significantly across regions. Thus, the state variable is a preferable choice of accounting for location. Since many states have relatively small sample sizes, a hierarchical model allows us to borrow information across states.

Comparing three full models with different grouping variables, the AIC and BIC score also suggest choosing `state` as the group-level variable.

Grouping	AIC	BIC
City	15408.58	15428.46
State	15354.88	15374.76
Region	15400.48	15420.36

Our baseline model incorporates only the state-level random intercepts. For other individual-level predictors, we add one variable to the model each time and use both the Likelihood Ratio test and the BIC score to determine whether it should be added. The **LRT** is designed for nested models while the BIC score considers

both the likelihood and the model complexity and gives a more general sense of model performance. The table below displays the results of model selection. We also used the full model as a starting point to perform stepwise backward elimination with the results agreeing the previous model selection method. (See appendix).

Our final model incorporates the grouping variable `state` and the individual level predictors `mgstr` (recoded as 4 levels), as well as `bulk_purchase`, `quarter`, and `source`.

Model	LRT.p.value	BIC
(1|state)		15374.76
(1|state) + mgstr2	0	14615.63
(1|state) + mgstr2 + bulk_purchase	2e-04	14610.01
(1|state) + mgstr2 + bulk_purchase + year	0.1079	14673.21
(1|state) + mgstr2 + bulk_purchase + quarter	0.0213	14626.19
(1|state) + mgstr2 + bulk_purchase + date_diff	0.1844	14616.87
(1|state) + mgstr2 + bulk_purchase + quarter + source	7e-04	14641.45
(1|state) + mgstr2 + bulk_purchase + quarter + source + primary_reason	1	14681.42

Interactions

To be added, all codes are in appendix, no interaction term can improve the model performance.

Final Model

Our final model is
$$log(y_{ij}) = \beta_0 + b_{0j} + \beta_1 M_{ij} + \beta_2 B_{ij} + \beta_3 Q_{ij} + \beta_4 S_{ij} + \epsilon_{ij}$$

$$b_{0j} \sim \mathcal{N}(0, \tau^2) \perp \epsilon_{ij} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$$

The response variable and predictors are defined as:

- $y_i j$: Per milligram price of morphine for individual i in state j

- M_{ij}: Dosage strength in mg of the units purchased, factored into 4 levels

- B_{ij}: Bulk purchase, an indicator for whether 10+ units were purchased at once

- Q_{ij}: Quarter of the reported purchase

- S_{ij}: Source of information (including first-hand and second-hand sources)

Model Diagnostics
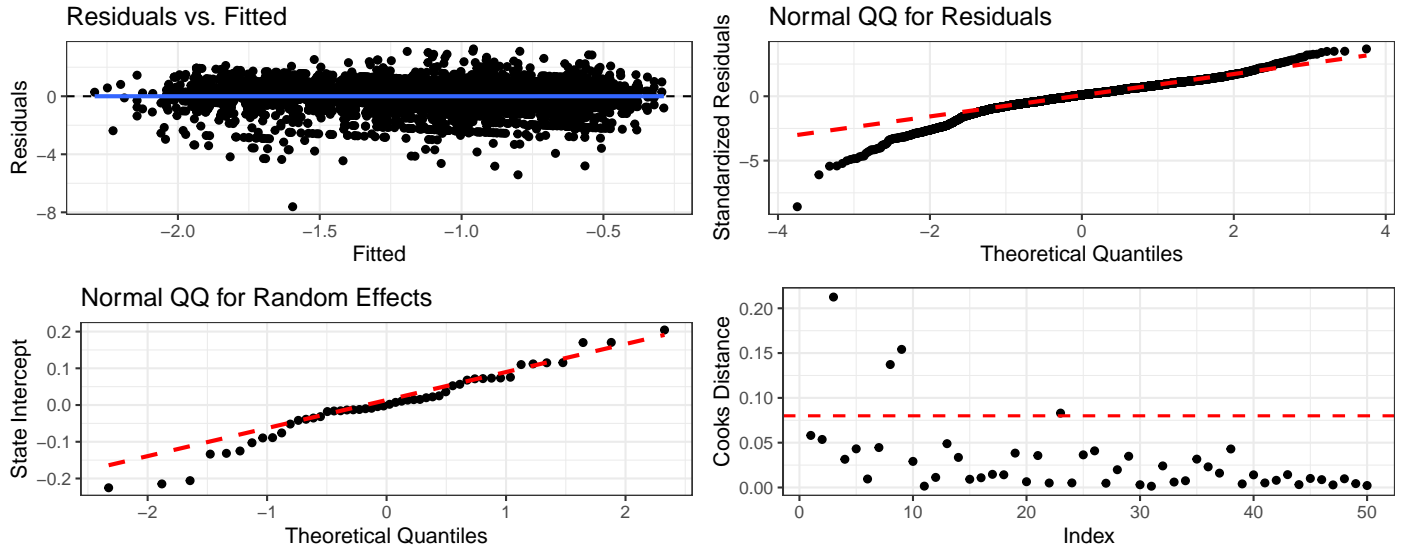
- `Residual vs. Fitted plot`: The residuals are spread equally around the horizontal line, indicating there is no non-linear relationship.
- `Normal QQ plot for residuals`: The normality assumption is slightly met since our residuals adhere around the diagonal line representing normality but have heavy tails on both sides. We also have one data point that deviates severely from the diagonal line.
- `Normal QQ plot for Random Effects`: We can accept the random effects are normally distributed. But we still have three outliers.
- `Cook's Distance`: We have 3 highly influential states (Florida, Pennsylvania, and California) whose Cook's distance exceeds the $\frac{4}{n}$ cutoff, where n denotes the number of states.

We tried to remove the data point with the lowest residual and the influential groups to address the violated assumptions. However, this did not drastically improve the normality of the residuals (see appendix). Moreover, the influential states have a considerable sample size (1382 observations). Therefore, we decide only to drop the individual level outlier but keep all the groups.
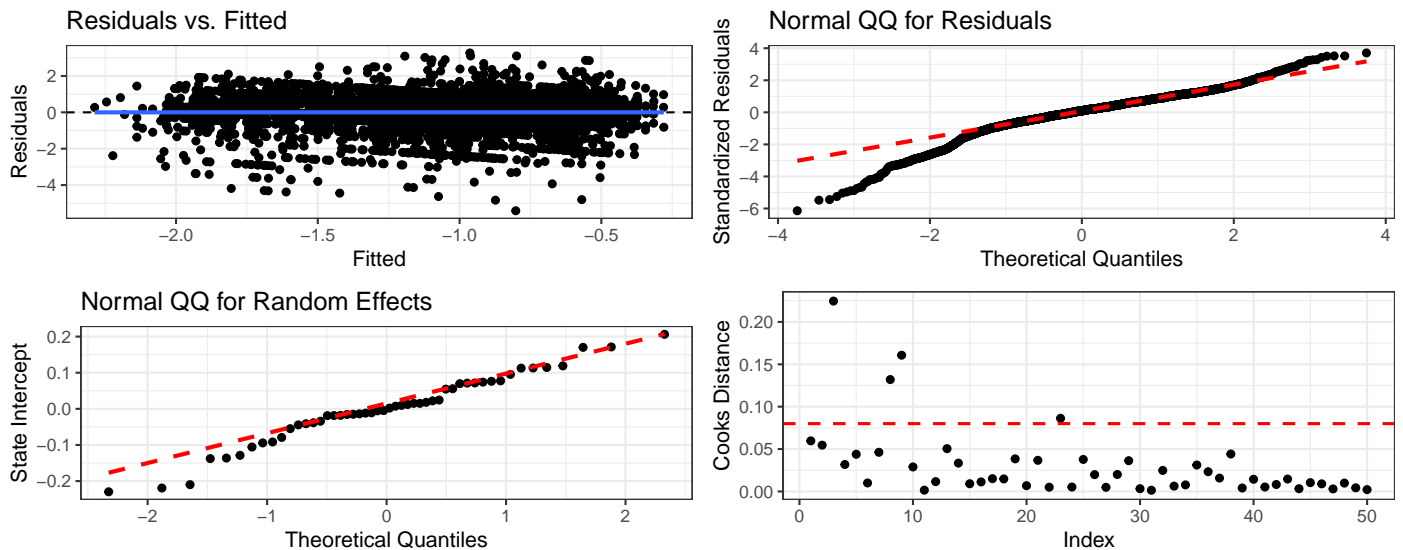
To be added, no significant improvement

Conclusion

Fixed Effects

	Estimate	exp(Estimate)	Std. Error	df	t value	Pr(>|t|)
(Intercept)	-0.6346	0.5301	0.0393	296.1819	-16.1290	0.0000
quarter2	0.0854	1.0891	0.0321	5550.5948	2.6578	0.0079
quarter3	0.0841	1.0877	0.0332	5555.0726	2.5349	0.0113
quarter4	0.0844	1.0881	0.0341	5551.1650	2.4759	0.0133
sourceHeard it	0.0633	1.0653	0.0335	5556.1197	1.8904	0.0588
sourceInternet	-0.0041	0.9959	0.0625	5555.5822	-0.0656	0.9477
sourceInternet Pharmacy	-0.3227	0.7242	0.1016	5548.9347	-3.1743	0.0015
sourcePersonal	-0.0398	0.9609	0.0281	5557.7473	-1.4157	0.1569
mgstr22 medium	-0.3816	0.6827	0.0279	5549.1951	-13.6631	0.0000
mgstr23 medium high	-0.7000	0.4966	0.0365	5554.7006	-19.1836	0.0000
mgstr24 high	-1.1197	0.3264	0.0420	5559.7107	-26.6889	0.0000
bulk_purchase1 Bulk purchase	-0.1141	0.8922	0.0296	5557.9880	-3.8488	0.0001

- Quarter (baseline: Quarter1): Compared with quarter 1, holding all other predictors unchanged, purchasing the morphine in quarter 2, the per milligram price of the drug will increase by a multiplicative effect of $e^{0.0853} = 1.0891$ (about 8.91%). Similarly, if the drug is purchased in quarter 3 or 4, the drug price will increase by 8.77% and 8.81%, respectively.

- Source (baseline: Blank): Compared with an unknown source, holding all other predictors unchanged, the per milligram drug price heard from other people will increase by a multiplicative effect of $e^{0.0633} = 1.0653$ (about 6.53%). Similarly, the price information obtained from the internet, internet pharmacy, or personal purchase will decrease by 0.41%, 27.58%, and 3.91%, respectively.

- Dosage Strength (baseline: Low): Compared with low dosage strength, holding all other predictors unchanged, the per milligram price of morphine will decrease by a multiplicative effect of $e^{-0.3816} = 0.6827$ (about 31.73%) if it has medium dosage strength. Similarly, if the dosage strength is medium-high or high, the drug price will decrease by 50.34% and 67.36%, respectively.

- Bulk Purchase (baseline: Not bulk purchase): Compared with non-bulk purchase, holding all other predictors unchanged, the unit price of morphine will decrease by a multiplicative effect of $e^{0.1141} = 0.8922$ (about 10.78%).

Random Effects

	τ^2	σ^2
Estimate	0.0161	0.7772

The estimated across-state variance is $\hat{\tau^2} = 0.0161$, which also describes the variation attributed to the random intercept. The estimated within-state variance is $\hat{\sigma^2} = 0.7772$, which describes the unexplained variation. The estimated interclass correlation is $\frac{\tau^2}{\tau^2+\sigma^2} \approx 0.02$. Therefore, we have little correlation within the same state.
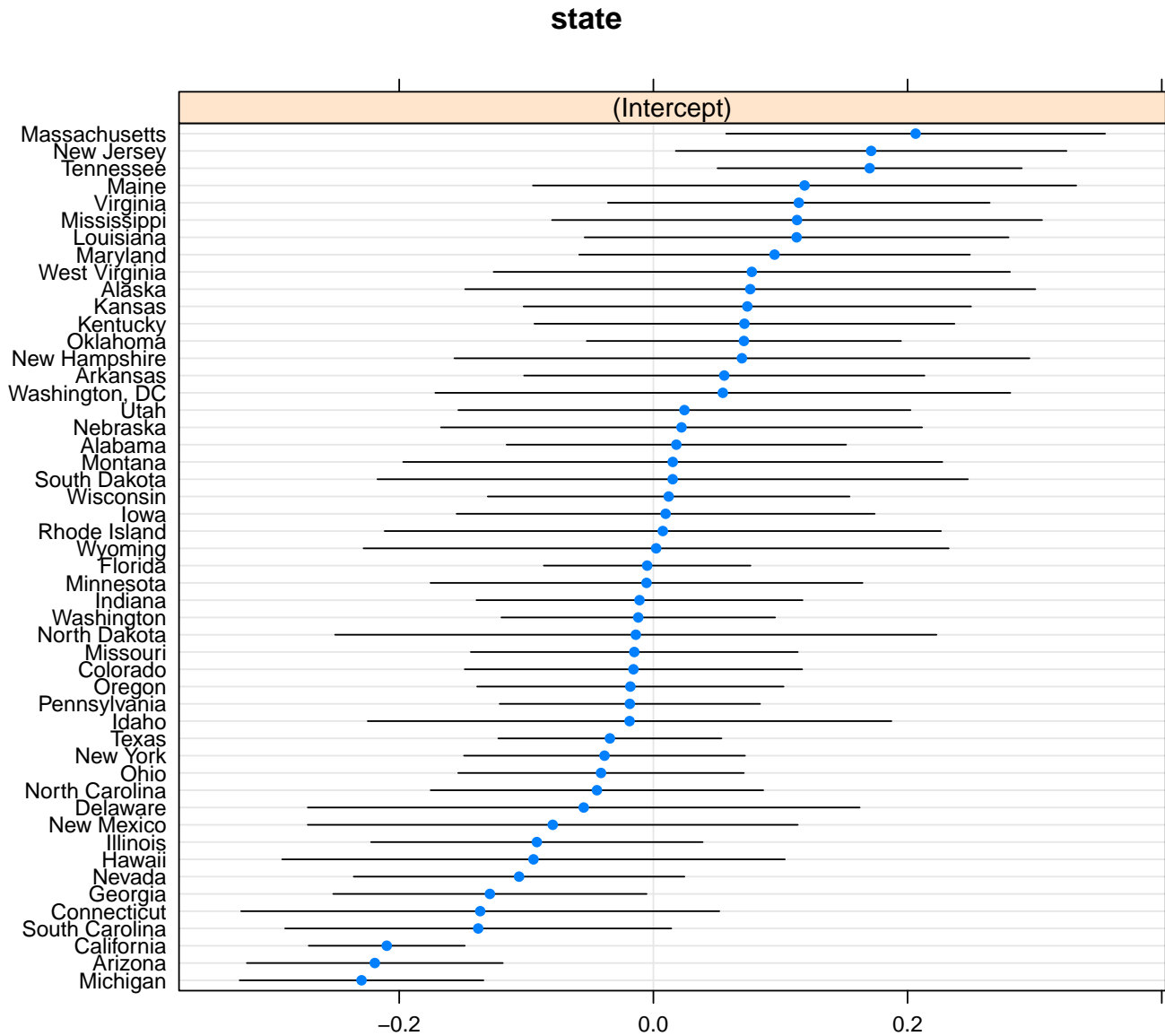
state

From the random intercepts plot, we can see that states have different bases per milligram morphine prices. The prices ranges from $e^{-0.2297} = 0.7948$ (Michigan) to $e^{0.2063} = 1.2292$ (Massachusetts). These estimates are based on the baseline condition of all other predictors, which are purchasing in quarter 1, from an unknown source, with low dosage strength, and not purchased in bulk

Limitation

Appendix

```r
# knitr::opts_chunk$set(warning=FALSE, message = FALSE, cache = TRUE)
library(tidyverse)
library(janitor)
library(gridExtra)
library(cowplot)
library(knitr)
require(magrittr)
require(dplyr)
library(kableExtra)
library(readr)
library(tidyr)
library(broom)
library(lme4)
library(glmmTMB)
library(sjPlot)
library(brms)
library(coda)
library(rstan)
library(tidybayes)
library(naniar)
library(olsrr)
library(lmerTest)
require(lattice)

# devtools::install_github("goodekat/redres")
library(redres)
```

```r
load('streetrx.RData')
```

```r
na_check <- streetrx %>%
  filter(api_temp == 'morphine') %>%
  mutate_all( list( ~na_if(., '') ) ) %>%
  droplevels()

dim(na_check)
sum(is.na(na_check))

sum(is.na(na_check$Primary_Reason))

sum(is.na(na_check$source))

sum(is.na(na_check))

gg_miss_upset(na_check)
```

```r
# subset for group drug

morph_data <- streetrx %>%
```

```r
    filter(api_temp == 'morphine')

morph_data$Primary_Reason <- droplevels(morph_data$Primary_Reason)
levels(morph_data$Primary_Reason)[1] <- "8 Prefer not to answer"
levels(morph_data$Primary_Reason)[2] <- "8 Prefer not to answer"

morph_data$source <- droplevels(morph_data$source)
levels(morph_data$source)[1] <- "Blank"

morph_data <- morph_data %>%
  filter(between(ppm, 0.000001, 10)) %>%
  mutate_all( list( ~na_if(., '') ) ) %>%
  drop_na() %>%
  clean_names() %>%
  mutate(
    quarter=substring(yq_pdate, 5, 5),
    year=substring(yq_pdate, 1, 4),
    state=recode_factor(droplevels(state), 'USA'='Unknown')
  )

nrow(morph_data)

sum(morph_data$ppm <=0)


# remove extreme outliers based on quantiles

# untransformed density
p1 <- morph_data %>%
  ggplot(aes(x=ppm)) +
    geom_histogram(
      aes(y=..density..),
      color='black',
      linetype='dashed',
      size=0.5,
      fill='lightblue',
      alpha=0.5,
      bins=20
    ) +
    geom_density(size=0.75, bw=0.3) +
    labs(title='Distribution of morphine ppm') +
    theme_bw()

# log-transformed density
p2 <- morph_data %>%
  ggplot(aes(x=log(ppm))) +
    geom_histogram(
      aes(y=..density..),
      color='black',
      linetype='dashed',
      size=0.5,
      fill='lightcoral',
```

```r
      alpha=0.5,
      bins=20
    ) +
    geom_density(size=0.75, bw=0.3) +
    labs(title='Distribution of morphine log(ppm)') +
    xlim(-7, 3) +
    theme_bw()

grid.arrange(p1, p2, ncol=2)


length(unique(morph_data$city))
length(unique(morph_data$state))
length(unique(morph_data$usa_region))


state_size <- morph_data %>%
  group_by(state) %>%
  summarise(n = n(), .groups = "drop")  %>%
  arrange(n) %>%
  pivot_wider(
    names_from=state,
    values_from=n
  )

state_size %>%
  dplyr::select(1:5) %>%
  kable(
    caption = '5 States with Smallest Sample Size',
    align='c',
    booktabs=TRUE) %>%
  kable_styling(latex_options = c('hold_position'))


# remove low sample size states
morph_data <- morph_data %>%
  mutate(state=as.character(state)) %>%
  filter(!state %in% c(
    'Puerto Rico', 'Vermont'
  ))


morph_state <- morph_data %>%
  filter(state %in% state.name) %>%
  mutate(state_abv=state.abb[match(state,state.name)])



grand_mean <- mean(morph_state$ppm)

p3 <- morph_state %>%
  group_by(state_abv) %>%
  summarise(n = n(), mean = mean(ppm)) %>%
  ggplot(aes(x=n, y=mean)) +
    geom_hline(
      aes(yintercept=grand_mean),
```

```
        linetype='dashed',
        color='red',
        size=0.75
    ) +
    geom_point() +
    labs(x='sample size', y='mean log(ppm)') +
    theme_bw()


p4 <- morph_state %>%
  ggplot(aes(y=log(ppm), x=state_abv)) +
    geom_boxplot(
      fill=rainbow(49),
      alpha=0.5
    ) +
    scale_x_discrete(guide=guide_axis(angle = 90)) +
    theme_bw() +
    labs(x='state')

cowplot::plot_grid(p3, p4, rel_widths = c(1, 2))
```

```
t1 <- morph_state %>%
  group_by(usa_region) %>%
  summarise(n=n(), mean=round(mean(log(ppm)), 3)) %>%
  tableGrob()

p5 <- morph_state %>%
  ggplot(aes(y=log(ppm), x=usa_region)) +
    geom_boxplot() +
    stat_summary(
      fun.y=mean,
      geom='point',
      color='red',
      size=3
    ) +
    theme_bw()

grid.arrange(t1, p5, ncol=2, widths=c(2, 2))
```

```
min(as.Date(morph_data$price_date, "%m/%d/%y")) #2013-01-01
morph_data %>% group_by(year) %>% summarise(n = n())
```

```
# remove data prior to 2010
morph_data <- morph_data %>%
  mutate(Year=as.character(year)) %>%
  filter(!year %in% c(
    1969, 2000, 2002, 2005
  ))
```

```r
# date_diff
morph_data <- morph_data %>%
  mutate(date_diff = as.numeric(
    as.Date(morph_data$price_date, "%m/%d/%y") - as.Date("2010-01-01")
    )
  )

morph_data %>%
  ggplot(aes(x=date_diff)) +
    geom_histogram(
      aes(y=..density..),
      color='black',
      linetype='dashed',
      size=0.5,
      fill='lightblue',
      alpha=0.5,
      bins=30
    ) +
    geom_density(size=0.75, bw=100) +
    labs(title='Date Distribution') +
    theme_bw()

morph_data %>%
  ggplot(aes(x=date_diff, y=log(ppm))) +
    geom_point() +
    geom_smooth() +
    theme_bw()

# check for random slopes
# morph_data_a <- subset(morph_data,state %in% c("Arizona", "Texas","California", "Pennsylvania"))
# morph_data_a %>% ggplot(aes(x = date_diff, y = log(ppm))) +
#   geom_point() +
#   geom_smooth() +
#   theme_bw() +
#   facet_wrap('state', scales = "fixed")

yearplot <- morph_data %>%
  ggplot(aes(x = year,y = log(ppm))) +
  geom_boxplot() +
  labs(x='Year') +
  stat_summary(
    fun.y=mean,
    geom='point',
    color='red',
    size=3
  ) +
  theme_bw()

quarterplot <- morph_data %>%
  ggplot(aes(x = quarter,y = log(ppm))) +
  geom_boxplot() +
```

```r
    labs(
      x='Quarter',
      y=''
    )  +
    stat_summary(
      fun.y=mean,
      geom='point',
      color='red',
      size=3
    ) +
    theme_bw()

grid.arrange(yearplot, quarterplot, ncol=2)
```

```r
morph_data %>%
  ggplot(aes(x = bulk_purchase,y = log(ppm))) +
  geom_boxplot() +
  labs(x='Bulk Purchase') +
  stat_summary(
    fun.y=mean,
    geom='point',
    color='red',
    size=3
  ) +
  theme_bw()
```

```r
# unique(morph_data$source)

# combine internet levels into single level
morph_data <- morph_data %>%
  mutate(source=replace(
    source, !source %in% c(
      "Blank",
      'Personal',
      'Heard it',
      'Internet',
      'Internet Pharmacy',
      'Drug forum'
    ), 'Internet'
  )) %>%
  droplevels()


morph_data <- morph_data %>%
  mutate(source=as.character(source)) %>%
  filter(source != "Drug forum")

morph_data %>%
  ggplot(aes(x = source,y = log(ppm))) +
  geom_boxplot() +
  stat_summary(
```

```r
      fun.y=mean,
      geom='point',
      color='red',
      size=2
  ) +
  theme_bw()



# morph_data %>%
#   group_by(source) %>%
#   summarize(n =n())
```

```r
morph_data %>%
  ggplot(aes(x = primary_reason,y =log(ppm))) +
  geom_boxplot() +
  coord_flip() +
  labs(x = "log(ppm)", y = "Reason")   +
    stat_summary(
      fun.y=mean,
      geom='point',
      color='red',
      size=3
    ) +
  theme_bw()
```

```r
morph_data %>%
  ggplot(aes(x=mgstr, y=log(ppm))) +
    geom_point() +
    geom_smooth() +
    theme_bw()

# morph_data %>%
#   ggplot(aes(x=log(mgstr), y=log(ppm))) +
#     geom_point() +
#     geom_smooth() +
#     theme_bw()

morph_data %>%
  ggplot(aes(x=mgstr)) +
    geom_histogram(
      aes(y=..density..),
      color='black',
      linetype='dashed',
      size=0.5,
      fill='lightblue',
      alpha=0.5,
      bins=10
    ) +
    geom_density(size=0.75, bw=7.5) +
    labs(title='mgstr Distribution') +
    theme_bw()
```

```r
# check for random slopes
morph_data %>% ggplot(aes(x = mgstr, y = log(ppm))) +
  geom_point() +
  geom_smooth() +
  theme_bw() +
  facet_wrap('usa_region', scales = "fixed")


morph_data %>%
  group_by(mgstr) %>%
  summarize(n = n()) %>%
  pivot_wider(
    names_from=mgstr,
    values_from=n
  ) %>%
  kable(
    caption='Sample Size for mgstr Levels',
    align='c',
    booktabs=TRUE
  ) %>%
  kable_styling(latex_options = c('hold_position'))

# inspect mgstr value quantiles
quantile(morph_data$mgstr, c(0.25, 0.5, 0.75)) %>%
  data.frame() %>%
  rename('mgstr'='.') %>%
  kable()

## here we decide to re-code mgstr by quantile
morph_data <- morph_data %>%
  mutate(mgstr2 = case_when(
    mgstr <= 15              ~ "1 low",
    mgstr >15 & mgstr <= 30  ~ "2 medium",
    mgstr >30 & mgstr <= 60  ~ "3 medium high",
    mgstr > 60               ~ "4 high")
  )

morph_data %>%
  ggplot(aes(x=mgstr2 ,y=log(ppm))) +
  geom_boxplot() +
  labs(y="log(ppm)", x="Strength")  +
  stat_summary(
    fun.y=mean,
    geom='point',
    color='red',
    size=3
  ) +
  theme_bw()

# jpeg("EDAplotStrength.jpg", width = 500, height = 300)
# dev.off()
```

```r
# group by city
mod_1 <- lmer(data=morph_data, log(ppm) ~ (1 |city), REML=F)

# group by state
mod_2 <- lmer(data=morph_data, log(ppm) ~ (1 |state), REML=F)

# group by region
mod_3 <- lmer(data=morph_data, log(ppm) ~ (1 |usa_region), REML=F)

aic_score <- sapply(c(mod_1, mod_2, mod_3), AIC)
bic_score <- sapply(c(mod_1, mod_2, mod_3), BIC)

data.frame('Grouping' = c('City', 'State', 'Region'), 'AIC' = aic_score, 'BIC' = bic_score) %>%
  kable()
```

```r
# appendix
# group by city
mod_full_1 <- lmer(data=morph_data, log(ppm) ~ (1 |city) + date_diff + quarter + year + mgstr2 +
                bulk_purchase + primary_reason + source, REML=F)

# group by state
mod_full_2 <- lmer(data=morph_data, log(ppm) ~ (1 |state) + date_diff + quarter + year + mgstr2 +
                bulk_purchase + primary_reason + source, REML=F)

# group by region
mod_full_3 <- lmer(data=morph_data, log(ppm) ~ (1 |usa_region) + date_diff + quarter + year + mgstr2
                bulk_purchase + primary_reason + source, REML=F)

aic_score <- sapply(c(mod_full_1, mod_full_2, mod_full_3), AIC)
bic_score <- sapply(c(mod_full_1, mod_full_2, mod_full_3), BIC)

data.frame('Grouping' = c('City', 'State', 'Region'), 'AIC' = aic_score, 'BIC' = bic_score) %>%
  kable()
```

```r
modela <- lmer(log(ppm) ~  (1|state), data = morph_data, REML=F) #
summary(modela)
```

```r
modela <- lmer(log(ppm) ~  (1|state), data = morph_data, REML=F) #



modelb <- lmer(log(ppm) ~  mgstr2 + (1|state), data = morph_data, REML=F) #




# =====
modelc <- lmer(log(ppm) ~  mgstr2 + bulk_purchase + (1|state), data = morph_data, REML=F) #

anova(modelb,modelc)
```

```r
# =====

modeld <- lmer(log(ppm) ~  year + mgstr2 + bulk_purchase + (1|state), data = morph_data, REML=F)

modele <- lmer(log(ppm) ~  quarter + mgstr2 + bulk_purchase + (1|state), data = morph_data, REML=F)

modelf <- lmer(log(ppm) ~ date_diff + mgstr2 + bulk_purchase + (1|state), data = morph_data, REML=F)

anova(modelc,modeld)

anova(modelc,modele)

anova(modelc,modelf)


# =====
modelg <- lmer(log(ppm) ~  quarter + source + mgstr2 + bulk_purchase + (1|state), data = morph_data,

anova(modele,modelg)

# =====

# Maybe a little cleaner way to do this
modelh<- lmer(log(ppm) ~  quarter + primary_reason + mgstr2 + bulk_purchase + (1|state), data = morp

anova(modelg, modelh)

modelAll <- lmer(log(ppm) ~ quarter + primary_reason + mgstr2 + bulk_purchase + year + date_diff + s
step(modelAll)
```

```r
round(anova(modelc,modele)$`Pr(>Chisq)`[2], 4)
```

```r
model <- c("(1|state)",
          "(1|state) + mgstr2",
          "(1|state) + mgstr2 + bulk_purchase",
          "(1|state) + mgstr2 + bulk_purchase + year",
          "(1|state) + mgstr2 + bulk_purchase + quarter",
          "(1|state) + mgstr2 + bulk_purchase + date_diff",
          "(1|state) + mgstr2 + bulk_purchase + quarter + source",
          "(1|state) + mgstr2 + bulk_purchase + quarter + source + primary_reason")
LRT <- c("",
        round(anova(modela,modelb)$`Pr(>Chisq)`[2],4),
        round(anova(modelb,modelc)$`Pr(>Chisq)`[2],4),
        round(anova(modelc,modeld)$`Pr(>Chisq)`[2],4),
        round(anova(modelc,modele)$`Pr(>Chisq)`[2],4),
        round(anova(modelc,modelf)$`Pr(>Chisq)`[2],4),
        round(anova(modele,modelg)$`Pr(>Chisq)`[2],4),
        round(anova(modelg,modelh)$`Pr(>Chisq)`[2],4))
BIC_score1 <- sapply(c(modela, modelb, modelc, modeld, modele, modelf, modelg, modelh), BIC)
data.frame("Model" = model, 'LRT p-value' = LRT, 'BIC' = BIC_score1) %>%
  kable()
```

```r
modelg <- lmer(log(ppm) ~  quarter + source + mgstr2 + bulk_purchase + (1|state), data = morph_data,

modelgg <- lmer(log(ppm) ~  quarter + source + mgstr2 + bulk_purchase + (1|state) +
                quarter * bulk_purchase, data = morph_data, REML=F)

anova(modelg,modelgg)

modelggg <- lmer(log(ppm) ~  quarter + source + mgstr2 + bulk_purchase + (1|state) +
                quarter * mgstr2, data = morph_data, REML=F)

anova(modelg,modelggg)

modelgggg <- lmer(log(ppm) ~  quarter + source + mgstr2 + bulk_purchase + (1|state) +
                bulk_purchase * mgstr2, data = morph_data, REML=F)

anova(modelg,modelgggg)

modelggggg <- lmer(log(ppm) ~  quarter + source + mgstr2 + bulk_purchase + (1|state) +
                quarter * source, data = morph_data , REML=F)

anova(modelg,modelggggg)

modelgggggg <- lmer(log(ppm) ~  quarter + source + mgstr2 + bulk_purchase + (1|state) +
                bulk_purchase * source, data = morph_data, REML=F)

anova(modelg,modelgggggg)


modelggggggg <- lmer(log(ppm) ~  quarter + source + mgstr2 + bulk_purchase + (1|state) +
                source * mgstr2, data = morph_data, REML=F)

anova(modelg,modelggggggg)
```

```r
modelg <- lmer(log(ppm) ~  quarter + source + mgstr2 + bulk_purchase + (1|state), data = morph_data,
```

```r
plot_qq <- function(model) {
  df <- data.frame(
    res=residuals(model, scaled=TRUE)
  )

  p <- ggplot(df, aes(sample=res)) +
    stat_qq() +
    stat_qq_line(
      linetype='dashed',
      color='red',
      size=1
    ) +
    labs(
      title='Normal QQ for Residuals',
      x='Theoretical Quantiles',
```

```r
      y='Standardized Residuals'
    ) + theme_bw()

  return(p)
}
plot_ranef_qq <- function(model) {
  df <- data.frame(
    res=ranef(model)[[1]][[1]]
  )

  p <- ggplot(df, aes(sample=res)) +
    stat_qq() +
    stat_qq_line(
      linetype='dashed',
      color='red',
      size=1
    ) +
    labs(
      title='Normal QQ for Random Effects',
      x='Theoretical Quantiles',
      y='State Intercept'
    ) + theme_bw()

  return(p)
}
plot_res_fit <- function(model) {
  df <- data.frame(
    res=residuals(model),
    fit=fitted(model)
  )

  p <- ggplot(df, aes(x=fit, y=res)) +
    geom_point() +
    geom_hline(
      yintercept=0,
      linetype="dashed"
    ) +
    geom_smooth() +
    labs(
      title='Residuals vs. Fitted',
      x='Fitted',
      y='Residuals'
    ) + theme_bw()

  return(p)
}
plot_scale_loc <- function(model) {
  df <- data.frame(
    res=sqrt(residuals(model, scaled=TRUE)),
    fit=fitted(model)
  )
```

```r
  p <- ggplot(df, aes(x=fit, y=res)) +
    geom_point() +
    geom_smooth() +
    labs(
      title='Scale-Location',
      x='Fitted',
      y=expression(sqrt('Standardized Residuals'))
    ) + theme_bw()

  return(p)
}
plot_res_dens <- function(model) {
  df <- data.frame(
    res=residuals(model)
  )

  p <- ggplot(df, aes(x=res)) +
    geom_density() +
    labs(
      title='Residuals Density',
      x='Residuals',
      y='Density'
    ) + theme_bw()

  return(p)
}


plot_cooks_distance <- function(model1){
  model_inf<- influence(model1, group = "state")
  data <- model.frame(model1)
  cooks_distance <- cooks.distance(model_inf)
  cutline <- 4 / length(unique(data$state))
  infindiv <- cooks_distance > cutline

  p <- ggplot(data=NULL, aes(x=1:length(unique(data$state)), y=cooks_distance)) +
    geom_point() +
    geom_hline(
      yintercept=cutline,
      linetype='dashed',
      color='red',
      size=0.75
    ) +
    labs(
      x='Index',
      y='Cooks Distance'
    ) +
    theme_bw()
  return(p)
}
```

```r
model_diag <- function(model) {
  p1 <- plot_res_fit(model)
  p2 <- plot_qq(model)
  p3 <- plot_ranef_qq(model)
  p4 <- plot_cooks_distance(model)

  cowplot::plot_grid(p1, p2, p3, p4, nrow=2)
}


# view_coef(modelg)
# view_params(modelg)
model_diag(modelg)


# remove lowest residual data point
morph_data2 <- morph_data[-which.min(resid(modelg)),]



model_g2 <- lmer(log(ppm) ~  quarter + source + mgstr2 + bulk_purchase + (1|state), data = morph_dat



# view_coef(model_g2)
# view_params(model_g2)
model_diag(model_g2)


model_g2_inf<- influence(model_g2, group = "state")


cooks_distance <- cooks.distance(model_g2_inf)
cutline <- 4 / length(unique(morph_data2$state))
infindiv <- cooks_distance > cutline

ggplot(data=NULL, aes(x=1:length(unique(morph_data2$state)), y=cooks_distance)) +
  geom_point() +
  geom_hline(
    yintercept=cutline,
    linetype='dashed',
    color='red',
    size=0.75
  ) +
  labs(
    x='Index',
    y='Cooks Distance'
  ) +
  theme_bw()


data.frame(
  rownames(model_g2_inf$`fixed.effects[-state]`),
  round(cooks_distance, 4),
  infindiv
) %>%
```

```r
  filter(infindiv == TRUE) %>%
  dplyr::select(1:2) %>%
  rename(`State`=1, `Cook's Distance`=2) %>%
  kable() %>%
  kable_classic(full_width=FALSE)

# remove three most influential states
morph_data3 <- morph_data2 %>%
  filter(!state %in% c('Florida', 'California','Pennsylvania'))



model_g3 <- lmer(log(ppm) ~  quarter + source + mgstr2 + bulk_purchase + (1|state), data = morph_dat

model_diag(model_g3)

# view coefficient estimates
view_coef <- function(model) {
  summary(model)$coefficients %>%
    as.data.frame() %>%
    mutate(`exp(Estimate)`=exp(Estimate)) %>%
    relocate(`exp(Estimate)`, .after=Estimate) %>%
    kable(digits = 4) %>%
    kable_classic(full_width=FALSE)
}




# view parameter estimates
view_params <- function(model) {
  params <- summary(model)$varcor %>%
    as.data.frame() %>%
    dplyr::select(vcov)

  rownames(params) <- c('$\\tau^2$', '$\\sigma^2$')
  colnames(params) <- c('Estimate')
  kable(params, digits = 4) %>%
    kable_classic(full_width=FALSE)
}

view_coef(model_g2)

view_params(model_g2)

# view intercept estimates and intervals
dotplot(ranef(model_g2, condVar = TRUE))

ranef(model_g2, condVar = TRUE) %>%
  as.data.frame() %>%
  arrange(desc(condval)) %>%
  mutate(`exp(condval)` = exp(condval)) %>%
  kable()
```