431 Class 09

Thomas E. Love, Ph.D.

2022-09-27

431

Today's Agenda

- Pulling in data for a new example, using `read_Rds()`

- Exploring a quantity, broken down into > 2 subgroups

  - Visualization gallery: comparison boxplot, faceted histograms, density and ridgeline plots

- Dealing with missing data via simple (single) imputation

- Using transformations to improve adherence to Normal assumptions, and Tukey's ladder of power transformations

Version 2022-09-24 17:35:28

431 CASE WESTERN RESERVE UNIVERSITY

Today's Setup

```
 1  knitr::opts_chunk$set(comment=NA)
 2  library(broom)                  ## tidy up model output
 3  library(equatiomatic)           ## pull equations from regressions
 4  library(ggrepel)                ## build useful labels in ggplot2
 5  library(ggridges)               ## help with ridgeline plots
 6  library(glue)                   ## work with strings
 7  library(kableExtra)             ## tidy up tables of output
 8  library(janitor)
 9  library(naniar)
10  library(simputation)
11  library(patchwork)
12  library(tidyverse)
13
14  theme_set(theme_bw())
```

Today's Data

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

```
1  bs_dat <- read_rds("c09/data/blood_storage.Rds")
```

- This allows us to read in the data just as they were last saved in R, including "factoring", etc.

  - `readRDS()` also works but is a little slower.

- To write an R data set, use `write_rds(datasetname, "locationoncomputer")`.

  - `saveRDS()` would also work, but slower.

The blood storage data set

This study[1] evaluates the association between red blood cells (RBC) storage duration (categorized into three groups) and time (in months) to biochemical prostate cancer recurrence after radical prostatectomy.

In cancer patients, perioperative blood transfusion has long been suspected of reducing long-term survival, and it is suspected that cancer recurrence may be worsened after the transfusion of older blood.

More complete versions of the data (along with more detailed explanations) appear in the Cleveland Clinic's Statistical Education repository, and in the `medicaldata` package in R.

1. Cata et al. "Blood Storage Duration and Biochemical Recurrence of Cancer after Radical Prostatectomy". *Mayo Clinic Proceedings* 2011; 86(2): 120-127. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3031436/

Codebook for **bs_dat** (n = 292 subjects)

Variable	Description
participant	subject identification code
age_group	younger, middle or older (RBC age exposure)
units	number of allogeneic blood transfusion units received
recur_time	time (months) to biochemical recurrence of prostate cancer

Our sample includes participants who received 1-4 units.

What's in the Data?

```
1  bs_dat
```

```
# A tibble: 292 × 4
   participant age_group units recur_time
   <chr>       <fct>     <dbl>      <dbl>
 1 102         older         2       47.6
 2 103         older         1       14.1
 3 104         middle        2       59.5
 4 105         middle        3        1.23
 5 106         older         1       74.7
 6 107         older         2       13.9
 7 108         younger       4        8.37
 8 109         younger       1       48.6
 9 110         middle        2       22.6
10 111         middle        2        4.63
# … with 282 more rows
```

Missing Values?

```
1  miss_var_summary(bs_dat)
```

```
# A tibble: 4 × 3
  variable    n_miss pct_miss
  <chr>        <int>    <dbl>
1 age_group        2    0.685
2 recur_time       1    0.342
3 participant      0    0
4 units            0    0
```
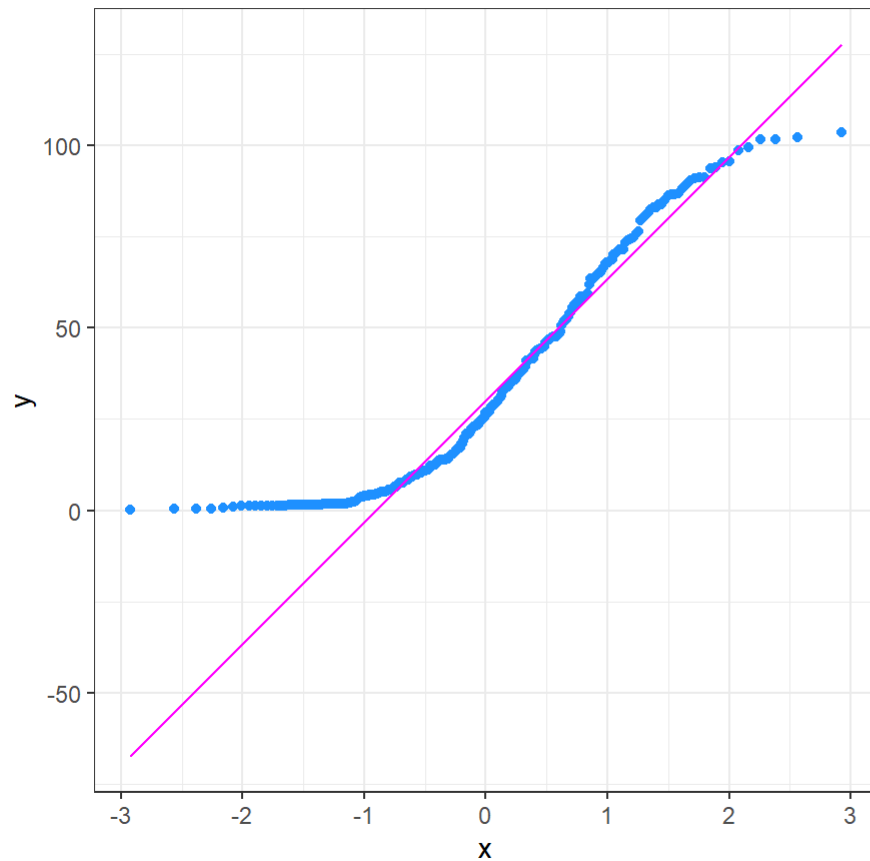
431 CASE WESTERN RESERVE UNIVERSITY

Outcome is time to recurrence

```r
1  p1 <- ggplot(bs_dat, aes(sample = recur_time)) +
2    geom_qq(col = "dodgerblue") +
3    geom_qq_line(col = "magenta") +
4    theme(aspect.ratio = 1) +
5    labs(title = "Normal Q-Q plot: recur_time")
6
7  p2 <- ggplot(bs_dat, aes(x = recur_time)) +
8    geom_histogram(aes(y = stat(density)),
9                   bins = 20, fill = "dodgerblue", col = "cyan") +
10   stat_function(fun = dnorm,
11                 args = list(mean = mean(bs_dat$recur_time, na.rm = TRUE),
12                             sd = sd(bs_dat$recur_time, na.rm = TRUE)),
13                 col = "magenta", lwd = 1.5) +
14   labs(title = "Density Function: recur_time")
15
16 p3 <- ggplot(bs_dat, aes(x = recur_time, y = "")) +
17   geom_boxplot(fill = "dodgerblue", notch = TRUE,
18                outlier.color = "dodgerblue") +
```

431
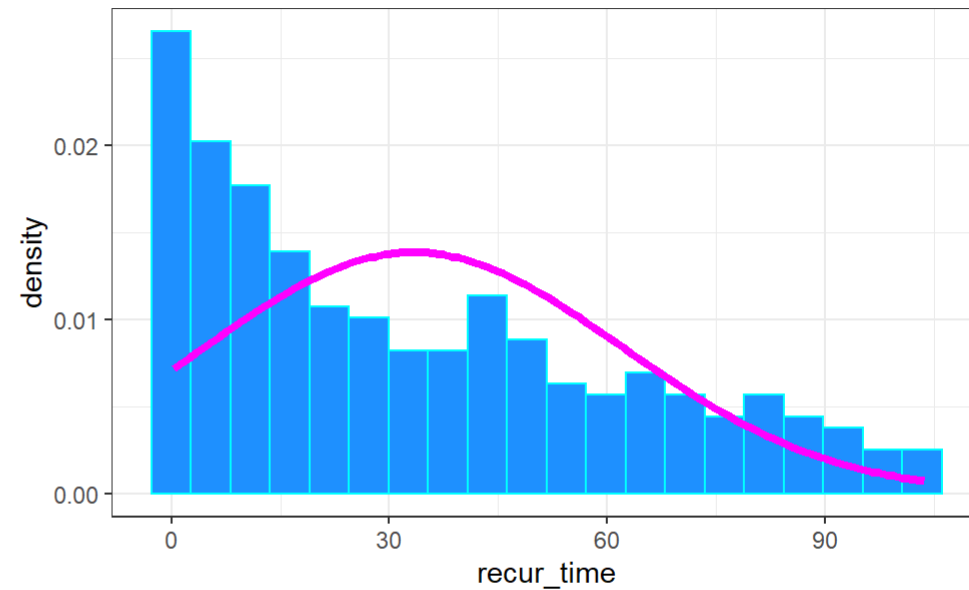
Outcome is time to recurrence
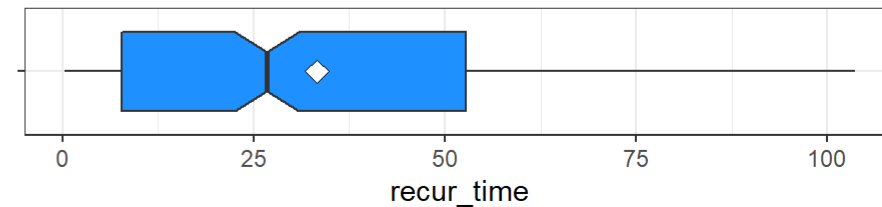
Compare `recur_time` by `age_group`

We'll start with a Complete Case Analysis that ignores any case with missing data.

```
1  bs_cc <- bs_dat |> filter(complete.cases(age_group, recur_time, units))
2
3  mosaic::favstats(recur_time ~ age_group, data = bs_cc) |>
4    kbl(digits = 2) |>
5    kable_styling(font_size = 28, full_width = FALSE)
```

age_group	min	Q1	median	Q3	max	mean	sd	n	missing
younger	0.27	9.28	31.18	52.27	101.7	34.29	29.75	96	0
middle	0.40	6.67	22.44	47.50	103.6	30.67	27.69	98	0
older	0.30	7.68	28.33	54.14	102.2	33.77	28.12	95	0

Scatterplot of `recur_time` vs. `age_group`

```
1  ggplot(bs_cc, aes(x = age_group, y = recur_time)) +
2    geom_point() + geom_smooth(method = "lm", se = FALSE)
```

Visualizing Strategies

We're trying to look at the impact of `age_group` on `recur_time`.

- Comparison Boxplot

- Faceted Histograms

- Overlapping Density Plot

- Ridgeline Plot

So let's walk through each of these.

431 CASE WESTERN RESERVE UNIVERSITY

Comparison Boxplot

```
1  ggplot(data = bs_cc, aes(x = age_group, y = recur_time)) +
2    geom_violin() +
3    geom_boxplot(aes(fill = age_group), width = 0.3,
4                 notch = TRUE, outlier.size = 2) +
5    guides(fill = "none") +
6    coord_flip() +
7    scale_fill_viridis_d(alpha = 0.5) +
8    labs(y = "Recurrence Time (in months)",
9         x = "Red Blood Cell age group",
10        title = "Recurrence Time by RBC Age Group")
```

Comparison Boxplot
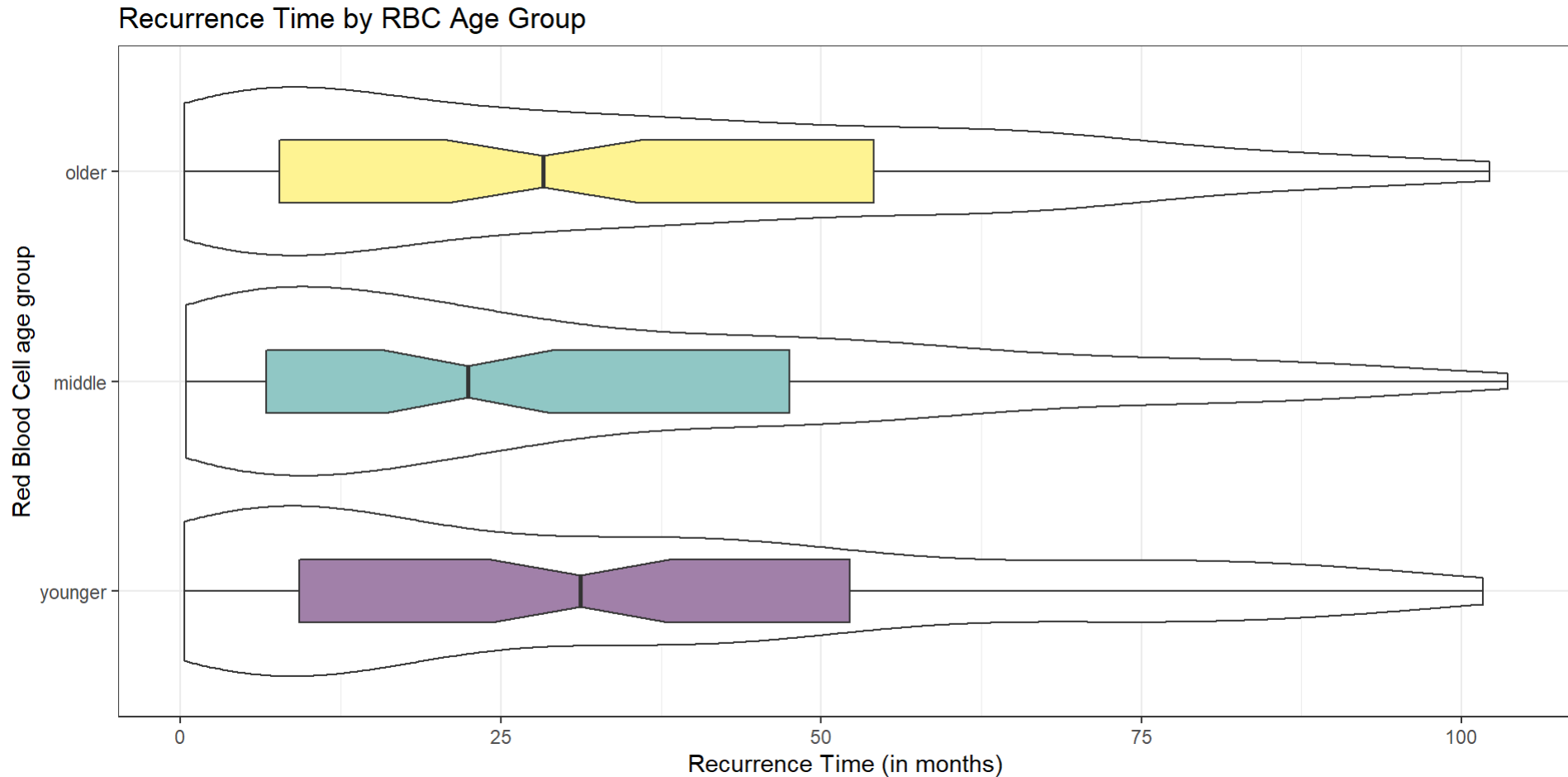
Recurrence Time by RBC Age Group

Add MEANS to Comparison Boxplot

```
1  ggplot(data = bs_cc, aes(x = age_group, y = recur_time)) +
2    geom_violin() +
3    geom_boxplot(aes(fill = age_group), width = 0.3,
4                 notch = TRUE, outlier.size = 2) +
5    stat_summary(fun = "mean", geom = "point",
6                 shape = 23, size = 3, fill = "white") +
7    guides(fill = "none") +
8    coord_flip() +
9    scale_fill_viridis_d(alpha = 0.5) +
10   labs(y = "Recurrence Time (in months)",
11        x = "Red Blood Cell age group",
12        title = "Recurrence Time by RBC Age Group",
13        caption = "Diamonds indicate sample means")
```

431

Add MEANS to Comparison Boxplot

Recurrence Time by RBC Age Group

Diamonds indicate sample means
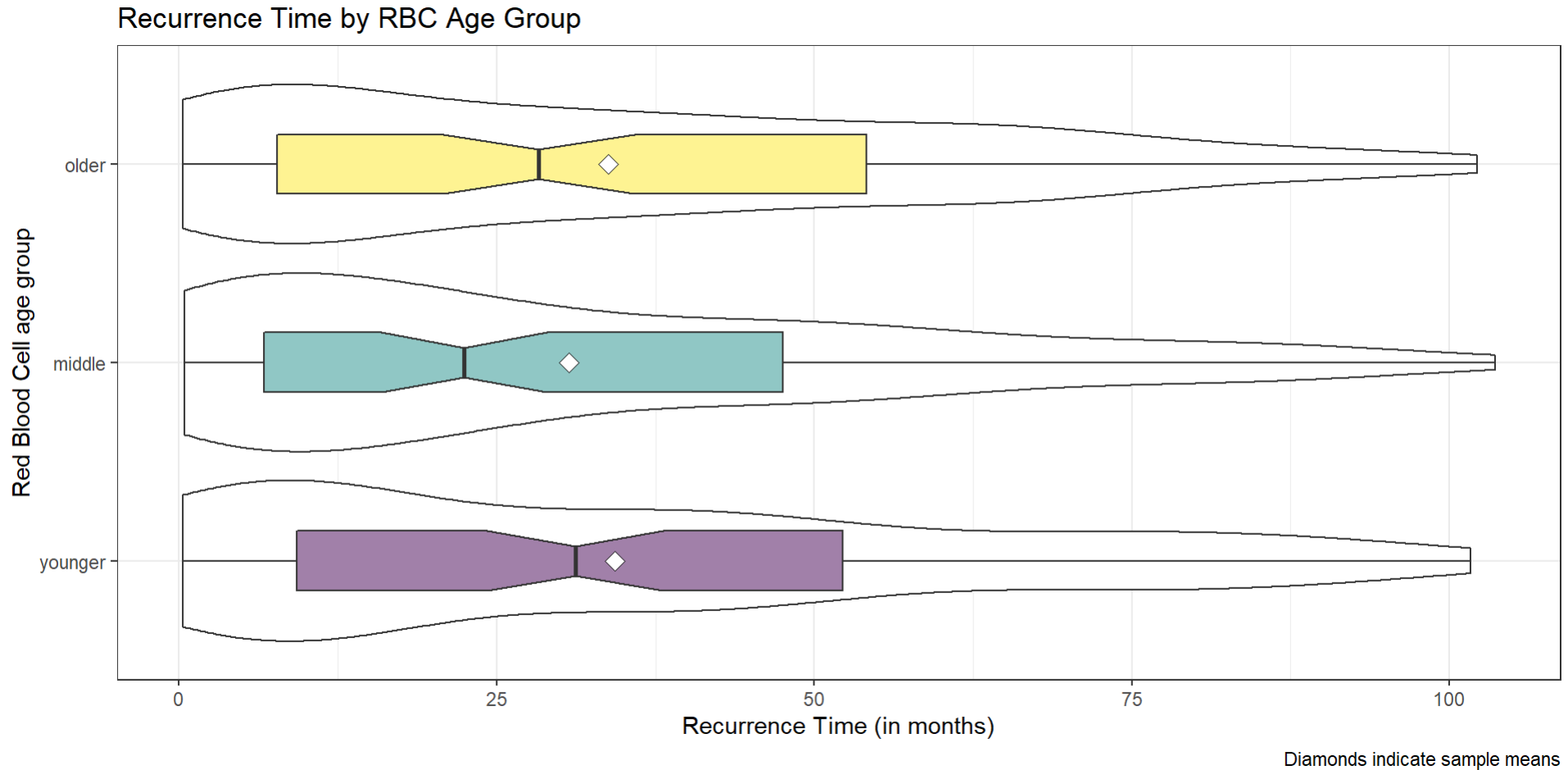
# Faceted Histograms

```
1  ggplot(data = bs_cc, aes(x = recur_time, fill = age_group)) +
2    geom_histogram(bins = 20, col = "navy") +
3    guides(fill = "none") +
4    facet_grid(age_group ~ .) +
5    labs(x = "Recurrence Time (in months)",
6         title = "Recurrence Time by RBC Age Group")
```

431 CASE WESTERN RESERVE UNIVERSITY

Faceted Histograms


Recurrence Time by RBC Age Group
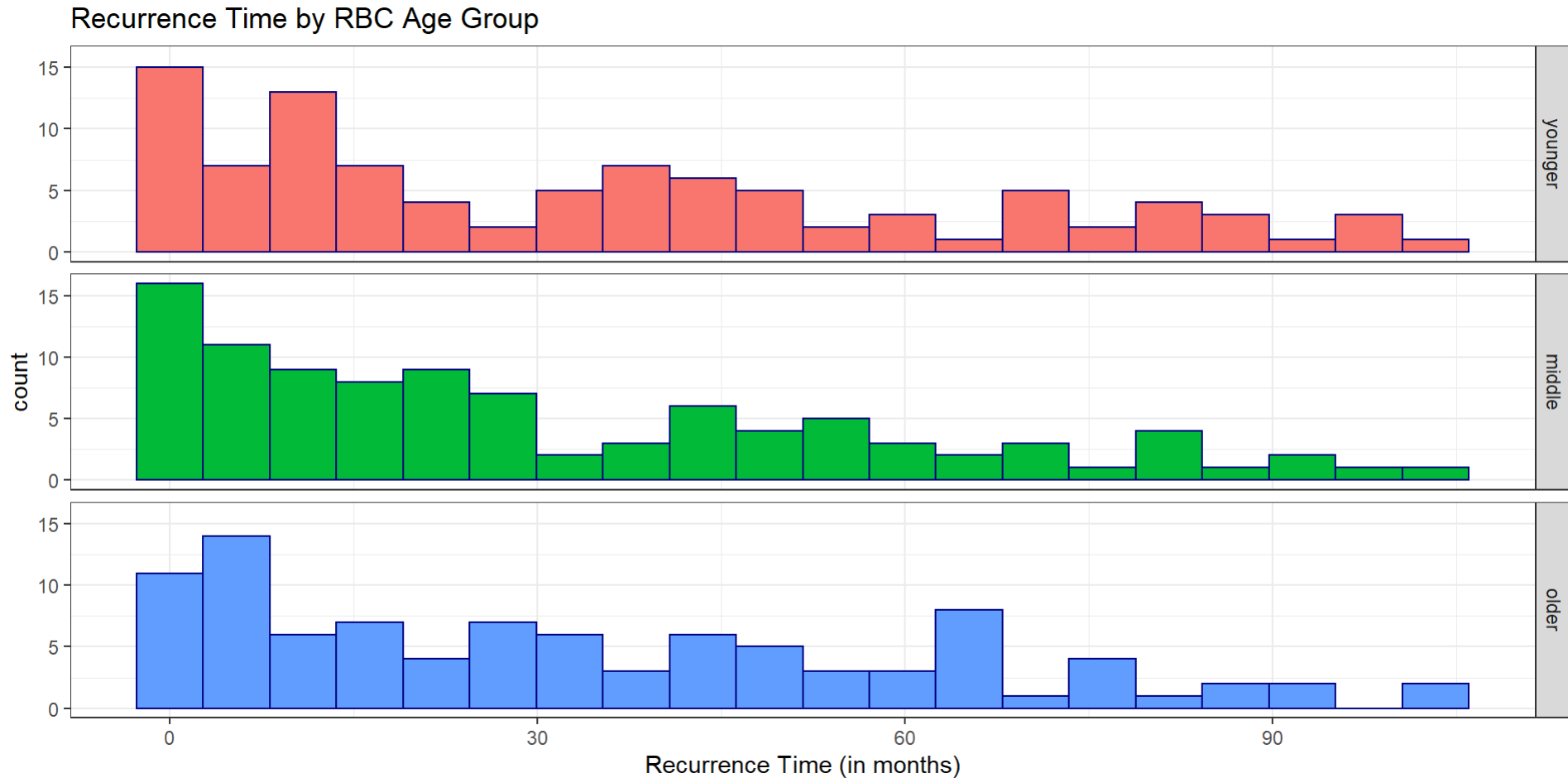
Comparing Densities

```
1  ggplot(data = bs_cc, aes(x = recur_time, fill = age_group)) +
2    geom_density() + scale_fill_viridis_d(alpha = 0.5, option = "A") +
3    labs(title = "Time to Recurrence, by RBC Age Group")
```


Time to Recurrence, by RBC Age Group

431
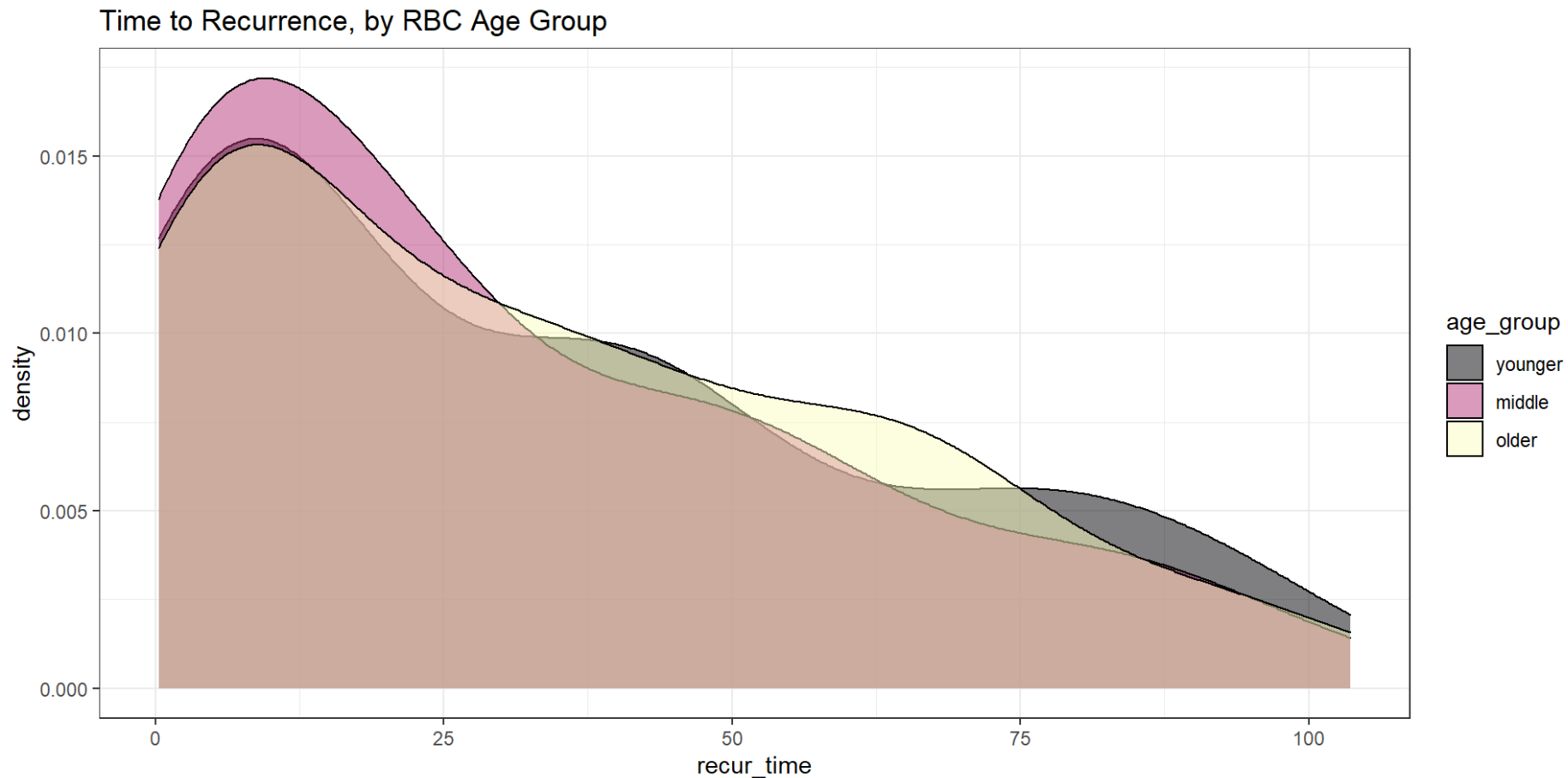
Using a Ridgeline Plot

```
1  ggplot(data = bs_cc, aes(x = recur_time, y = age_group,
2                          fill = age_group)) +
3    geom_density_ridges(alpha = 0.5) +
4    guides(fill = "none") +
5    labs(title = "Time to Recurrence, by RBC Age Group")
```


Time to Recurrence, by RBC Age Group
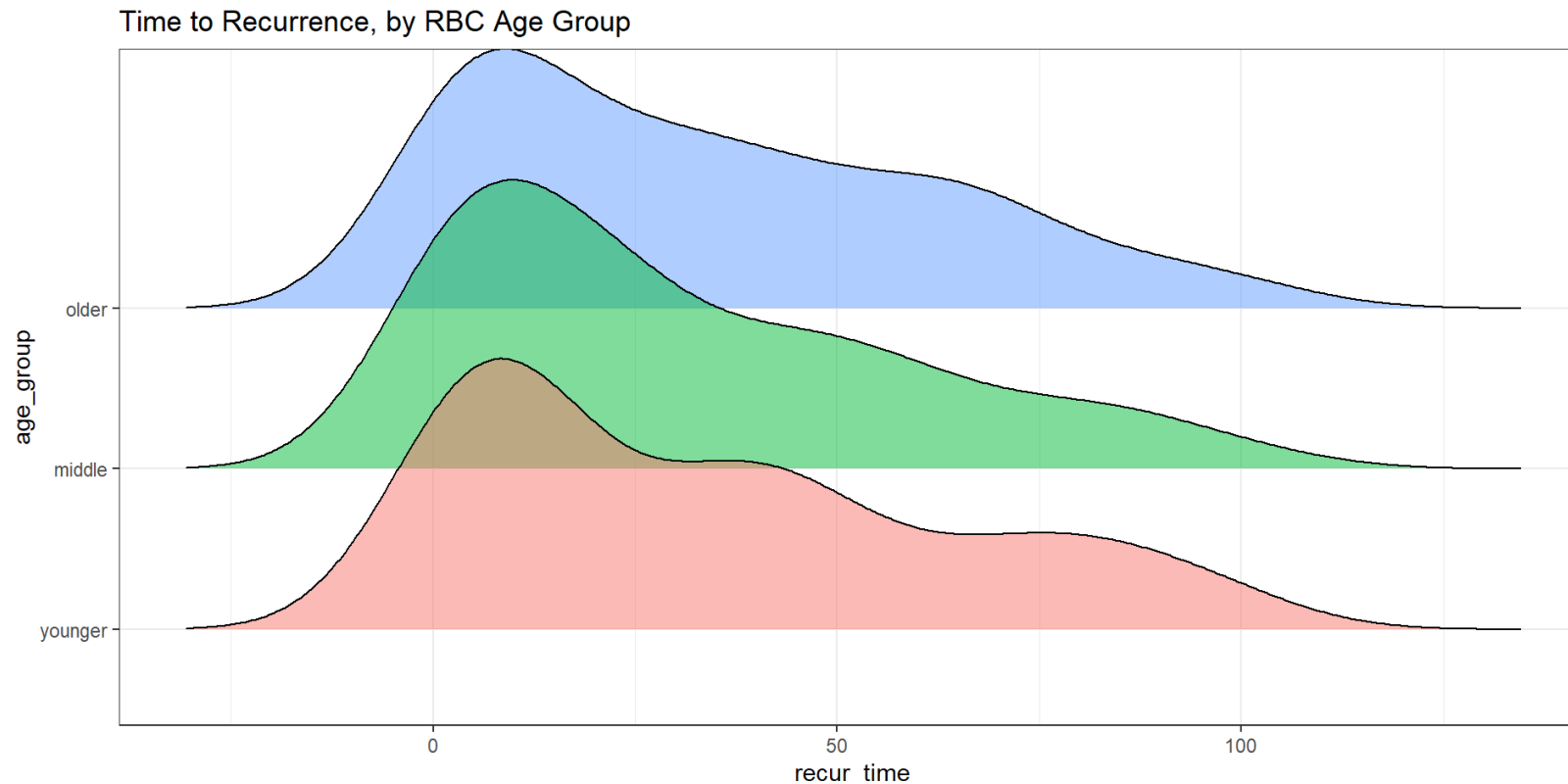
Complete Cases: Model Time using Age

```
1  m1 <- lm(recur_time ~ age_group, data = bs_cc)
2
3  m1
```

```
Call:
lm(formula = recur_time ~ age_group, data = bs_cc)

Coefficients:
    (Intercept)    age_groupmiddle     age_groupolder
        34.2885            -3.6143            -0.5193
```

431 | CASE WESTERN RESERVE UNIVERSITY

Extract Equation with **equatiomatic**

```
1  extract_eq(m1, use_coefs = TRUE, wrap = TRUE, coef_digits = 2,
2              terms_per_line = 1, operator_location = "start",
3              font_size = "small")
```

$$\widehat{recur\_time} = 34.29$$
$$- 3.61(age\_group_{middle})$$
$$- 0.52(age\_group_{older})$$

431 CASE WESTERN RESERVE UNIVERSITY

$$\widehat{\text{recur\_time}} = 34.29 - 3.61(\text{age\_group}_{\text{middle}}) \\ - 0.52(\text{age\_group}_{\text{older}})$$

age_group	m1 estimate of recur_time (months)
Younger	
Middle	
Older	

$$\widehat{\text{recur\_time}} = 34.29 - 3.61(\text{age\_group}_{\text{middle}})$$
$$- 0.52(\text{age\_group}_{\text{older}})$$

age_group	m1 estimate of recur_time (months)
Younger	34.29
Middle	
Older	

431 CASE WESTERN RESERVE UNIVERSITY

$$\widehat{\text{recur\_time}} = 34.29 - 3.61(\text{age\_group}_{\text{middle}})$$
$$- 0.52(\text{age\_group}_{\text{older}})$$

age_group	m1 estimate of recur_time (months)
Younger	34.29
Middle	34.29 - 3.61 = 30.68
Older	

431 CASE WESTERN RESERVE UNIVERSITY

$$\widehat{\text{recur\_time}} = 34.29 - 3.61(\text{age\_group}_{\text{middle}})$$
$$- 0.52(\text{age\_group}_{\text{older}})$$

age_group	m1 estimate of recur_time (months)
Younger	34.29
Middle	34.29 - 3.61 = 30.68
Older	34.29 - 0.52 = 33.77

431 CASE WESTERN RESERVE UNIVERSITY

Sample Means from `bs_cc`

```
1  mosaic::favstats(recur_time ~ age_group, data = bs_cc) |>
2    select(age_group, mean) |>
3    kbl(digits = 2) |> kable_styling(font_size = 28)
```

age_group	mean
younger	34.29
middle	30.67
older	33.77

Compare to m1 estimates (some rounding)

age_group	Younger	Middle	Older
Est. recur_time	34.29	30.68	33.77

Tidy coefficients with **broom** package

```
1  tidy(m1, conf.int = TRUE, conf.level = 0.90) |>
2    kbl(digits = 2) |> kable_styling(font_size = 28)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	34.29	2.91	11.78	0.00	29.48	39.09
age_groupmiddle	-3.61	4.10	-0.88	0.38	-10.38	3.15
age_groupolder	-0.52	4.13	-0.13	0.90	-7.33	6.29

- What is the 90% CI for the population mean time to recurrence for `age_group` = Younger?

- What is the 90% CI for the mean difference in time to recurrence between Younger and Middle?

431

glance to summarize m1's fit

- The broom package has three main functions, tidy(), glance() and augment()

```
1  glance(m1) |>
2    select(r.squared, AIC, nobs, df, df.residual) |>
3    kbl(digits = c(4, 1, 0, 0, 0)) |> kable_styling(font_size = 28)
```

r.squared	AIC	nobs	df	df.residual
0.0032	2762	289	2	286

Imputation

431

Dealing with the Missing Data

We have done all analyses on complete cases, but that's not always wise.

- What if doing so would bias our conclusions?

- Here we have two missing `age_group` values and one missing `recur_time`.

It's scary to estimate these missing values. What could we do?

Single Imputation

In single imputation analyses, NA values are estimated/replaced one time with one particular data value for the purpose of obtaining more complete samples, at the expense of creating some potential bias in the eventual conclusions or obtaining slightly less accurate estimates than would be available if there were no missing values in the data.

- The `simputation` package can help us execute single imputations using a wide variety of techniques, within the pipe approach used by the tidyverse.

See Section 9.8 of the Course Notes for some additional examples.

Estimate missing values?

```
1  bs_dat |> select(-participant) |> summary()
```

```
  age_group            units              recur_time
younger:97   Min.    :1.000    Min.    :  0.270
middle :98   1st Qu.:2.000    1st Qu.:  7.685
older  :95   Median :2.000    Median : 26.690
NA's   : 2   Mean   :2.048    Mean   : 33.297
             3rd Qu.:2.000    3rd Qu.: 52.685
             Max.   :4.000    Max.   :103.600
                              NA's   :1
```

Which values are missing and must be imputed?

431 CASE WESTERN RESERVE UNIVERSITY

Create an imputation model

The `simputation` package is our friend here. We'll use

- `impute_pmm()` to impute quantities, and

- `impute_cart()` to impute factors, for now.

```
1  bs_imp <- bs_dat |>
2    impute_pmm(recur_time ~ age_group + units) |>
3    impute_cart(age_group ~ units)
```

We start with no missing `units` so we use that to impute `age_group`, then use both `age_group` and `units` to impute `recur_time`. Any missing data now?

Compare Results

```
1  summary(bs_dat)
```

```
 participant          age_group        units            recur_time
 Length:292           younger:97    Min.   :1.000    Min.   :  0.270
 Class :character     middle :98    1st Qu.:2.000    1st Qu.:  7.685
 Mode  :character     older  :95    Median :2.000    Median : 26.690
                      NA's   : 2    Mean   :2.048    Mean   : 33.297
                                    3rd Qu.:2.000    3rd Qu.: 52.685
                                    Max.   :4.000    Max.   :103.600
                                                     NA's   :1
```

```
1  summary(bs_imp)
```

```
 participant          age_group        units            recur_time
 Length:292           younger:98    Min.   :1.000    Min.   :  0.270
 Class :character     middle :98    1st Qu.:2.000    1st Qu.:  7.728
 Mode  :character     older  :96    Median :2.000    Median : 26.695
                                    Mean   :2.048    Mean   : 33.301
                                    3rd Qu.:2.000    3rd Qu.: 52.492
                                    Max.   :4.000    Max.   :103.600
```

Model Time Using Age with `bs_imp`

```
1  m1_imp <- lm(recur_time ~ age_group, data = bs_imp)
2
3  extract_eq(m1_imp, use_coefs = TRUE, wrap = TRUE, coef_digits = 2,
4            terms_per_line = 1, operator_location = "start",
5            font_size = "small")
```

$$\widehat{\text{recur\_time}} = 34.85$$
$$- 4.18(\text{age\_group}_{\text{middle}})$$
$$- 0.46(\text{age\_group}_{\text{older}})$$

Compare Tidied Coefficients

```
1  tidy(m1, conf.int = TRUE, conf.level = 0.90) |>
2    kbl(digits = 2) |> kable_styling(font_size = 28)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	34.29	2.91	11.78	0.00	29.48	39.09
age_groupmiddle	-3.61	4.10	-0.88	0.38	-10.38	3.15
age_groupolder	-0.52	4.13	-0.13	0.90	-7.33	6.29

```
1  tidy(m1_imp, conf.int = TRUE, conf.level = 0.90) |>
2    kbl(digits = 2) |> kable_styling(font_size = 28)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	34.85	2.91	11.99	0.00	30.06	39.65
age_groupmiddle	-4.18	4.11	-1.02	0.31	-10.97	2.60
age_groupolder	-0.46	4.13	-0.11	0.91	-7.28	6.36

431

Compare Summaries with `glance`

```
1  glance(m1) |>
2    select(r.squared, AIC, nobs, df, df.residual) |>
3    kbl(digits = c(4, 1, 0, 0, 0)) |> kable_styling(font_size = 28)
```

r.squared	AIC	nobs	df	df.residual
0.0032	2762	289	2	286

```
1  glance(m1_imp) |>
2    select(r.squared, AIC, nobs, df, df.residual) |>
3    kbl(digits = c(4, 1, 0, 0, 0)) |> kable_styling(font_size = 28)
```

r.squared	AIC	nobs	df	df.residual
0.0043	2795.7	292	2	289

431

What Type of Missingness do we have?

1. MCAR = Missingness completely at random.

A variable is missing completely at random if the probability of missingness is the same for all units, for example, if for each subject, we decide whether to collect data on a measure by rolling a die and refusing to answer if a "6" shows up. If data are missing completely at random, then throwing out cases with missing data (i.e. doing a complete case analysis) does not bias your inferences.

431 CASE WESTERN RESERVE UNIVERSITY

What Type of Missingness do we have?

2. MAR = Missingness at random.

Missingness that depends only on observed predictors. A more general assumption, called missing at random or MAR, is that the probability a variable is missing depends only on available information. Here, we would have to be willing to assume that the probability of nonresponse to depends only on the other, fully recorded variables in the data.

- Here is the situation that most obviously cries out for imputation.

431 CASE WESTERN RESERVE UNIVERSITY

What Type of Missingness do we have?

3. Missing not at random

This is a bigger problem, and includes both:

- Missingness that depends on unobserved predictors. Missingness is no longer "at random" if it depends on information that has not been recorded and this information also predicts the missing values.

- Missingness that depends on the missing value itself. For example, suppose that people with higher earnings are less likely to reveal them.

OK, back to our Model m1 with complete cases for the rest of today...

431

Save residuals and fitted values for m1

```
1  m1_aug <- augment(m1, data = bs_cc)
2
3  m1_aug
```

```
# A tibble: 289 × 10
   particip…¹ age_g…² units recur…³ .fitted .resid   .hat .sigma .cooksd
.std.…⁴
   <chr>      <fct>   <dbl>   <dbl>   <dbl>  <dbl>  <dbl>  <dbl>   <dbl>
<dbl>
 1 102        older       2    47.6    33.8   13.9 0.0105   28.6 8.46e-4
0.488
 2 103        older       1    14.1    33.8  -19.7 0.0105   28.6 1.70e-3
-0.693
 3 104        middle      2    59.5    30.7   28.8 0.0102   28.5 3.54e-3
1.01
 4 105        middle      3     1.23   30.7  -29.4 0.0102   28.5 3.70e-3
-1.04
 5 106        older       1    74.7    33.8   40.9 0.0105   28.5 7.38e-3
1.44
 6 107        older       2    13.0    33.8  -19.9 0.0105   28.6 1.74e-3
```

m1 Residuals vs. Fitted Values

```
1  ggplot(data = m1_aug, aes(x = .fitted, y = .resid)) +
2    geom_point()
```
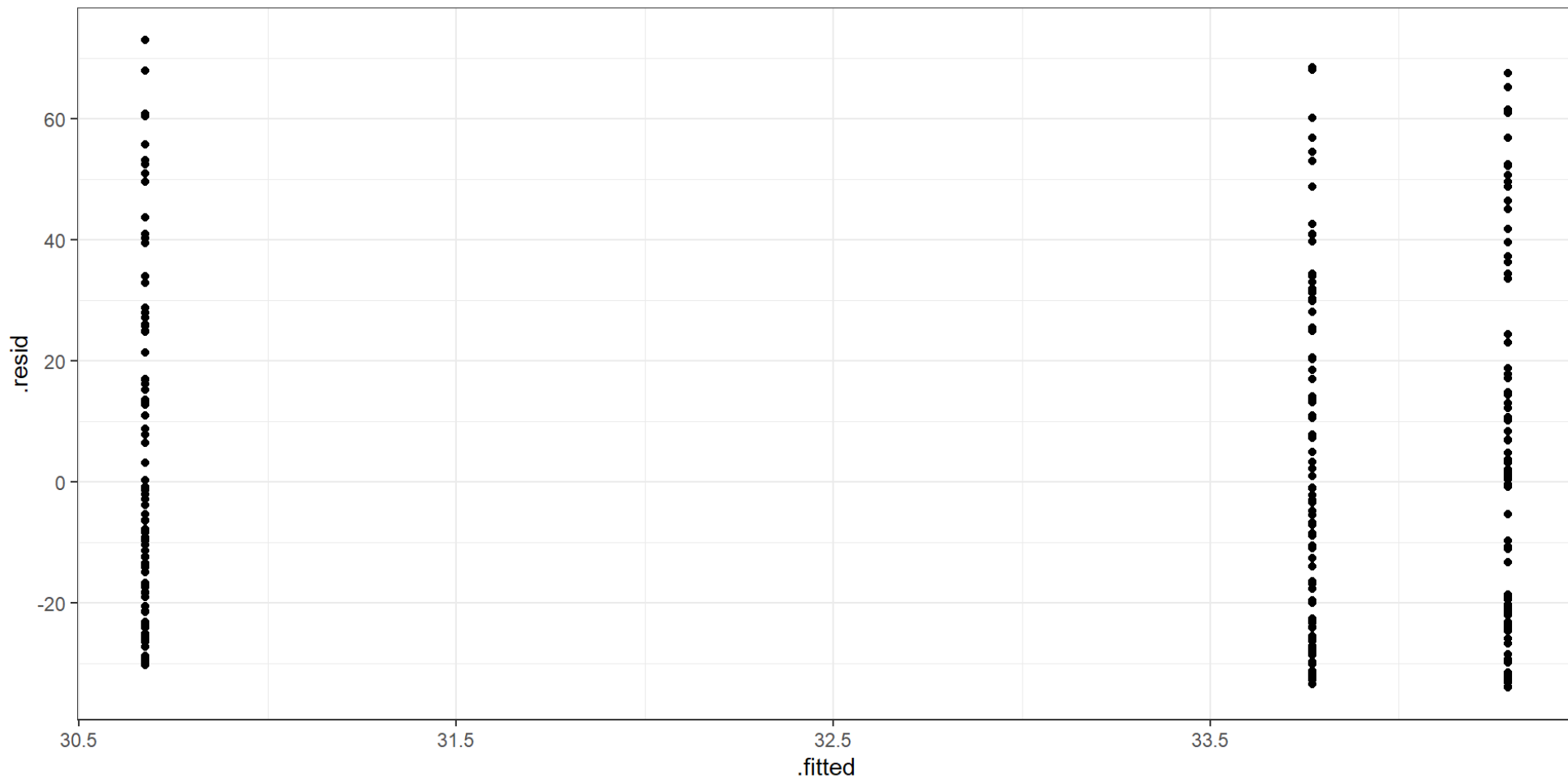
Normal Q-Q plot of m1 Residuals

```
1  ggplot(data = m1_aug, aes(sample = .resid)) +
2    geom_qq() + geom_qq_line(col = "red")
```
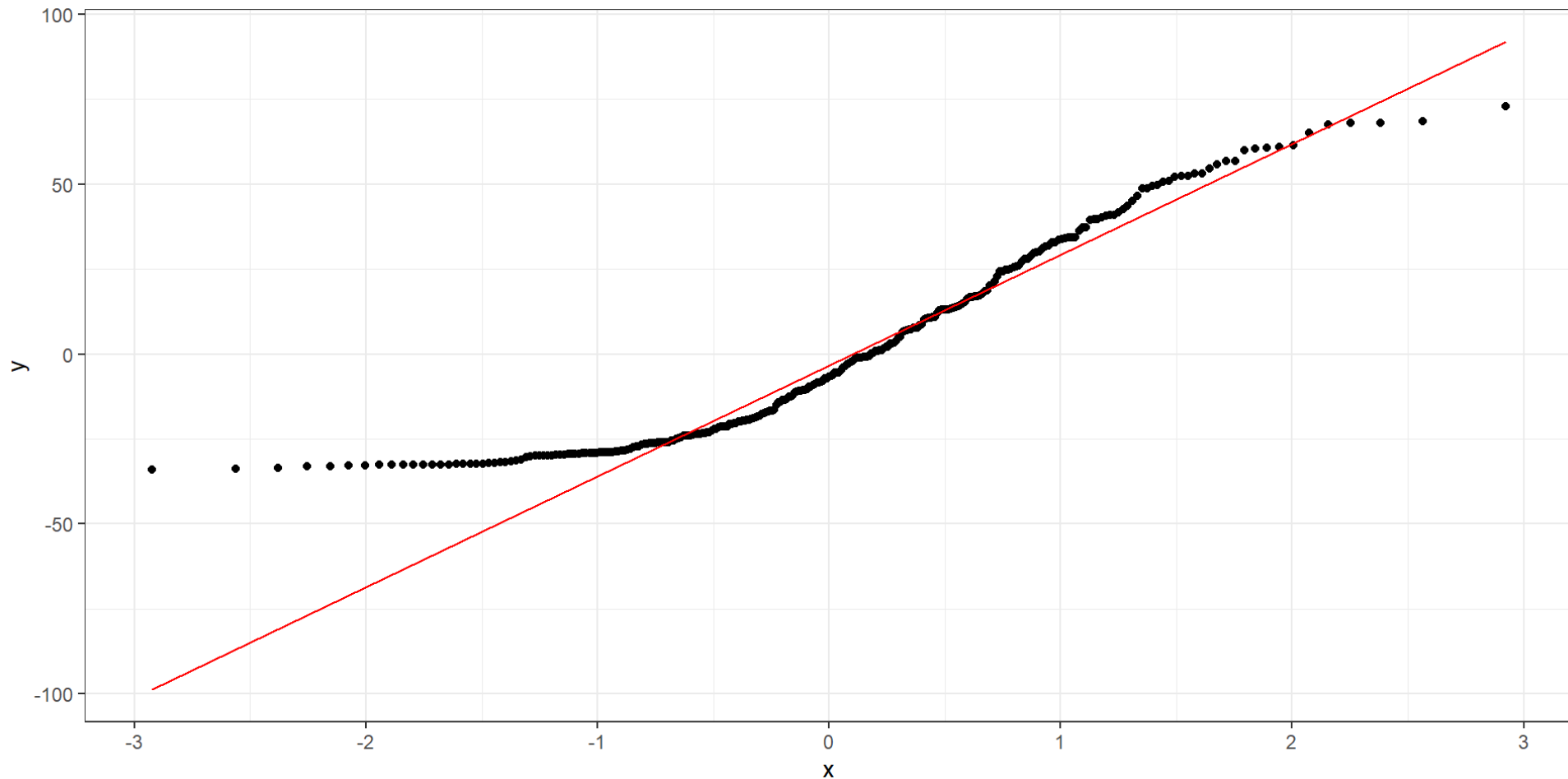
Back to our Comparison Boxplot

- Does comparing means make sense here?

- Are these sample distributions "Normal-ish"?

Recurrence Time by RBC Age Group
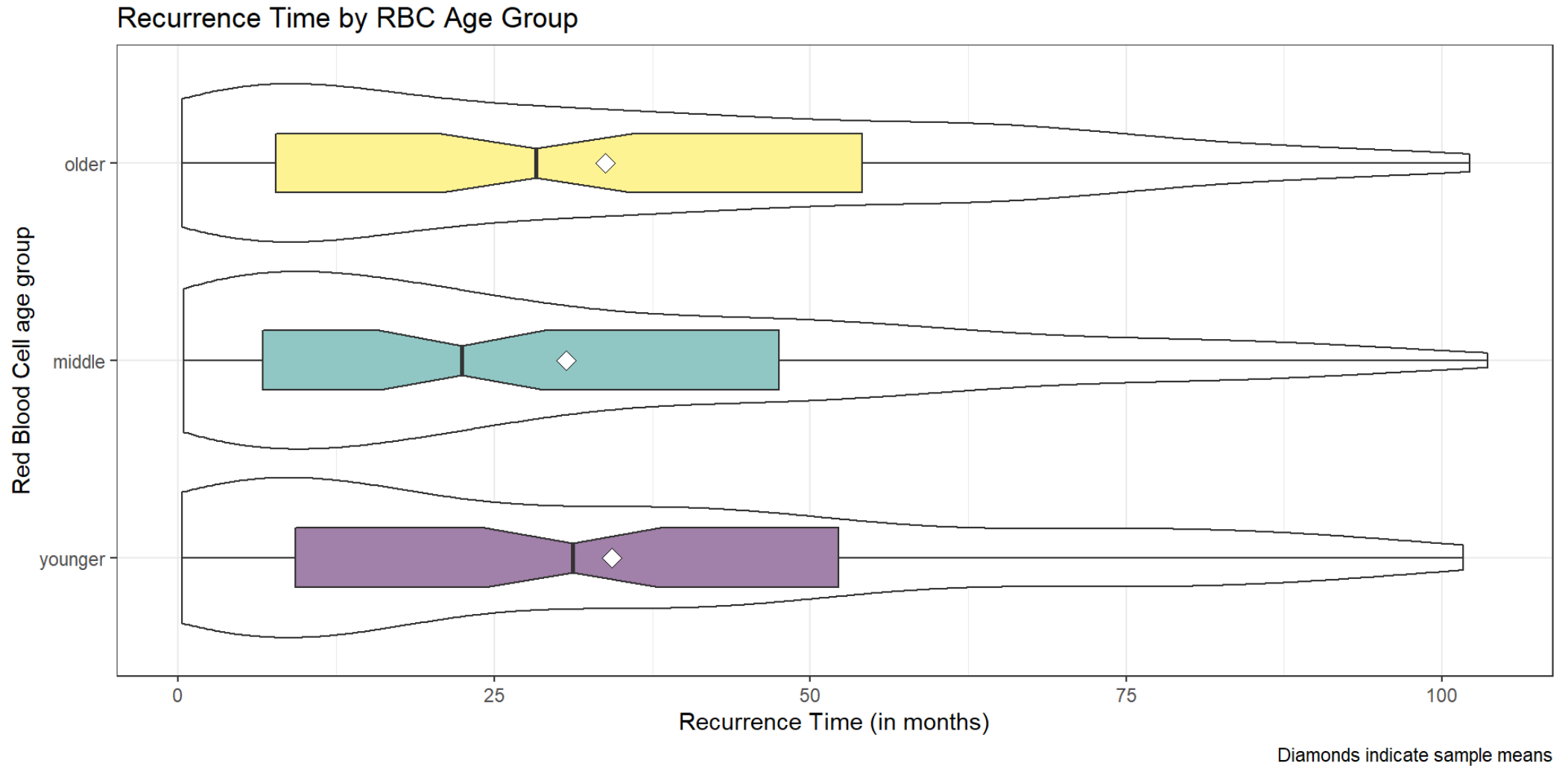
Diamonds indicate sample means
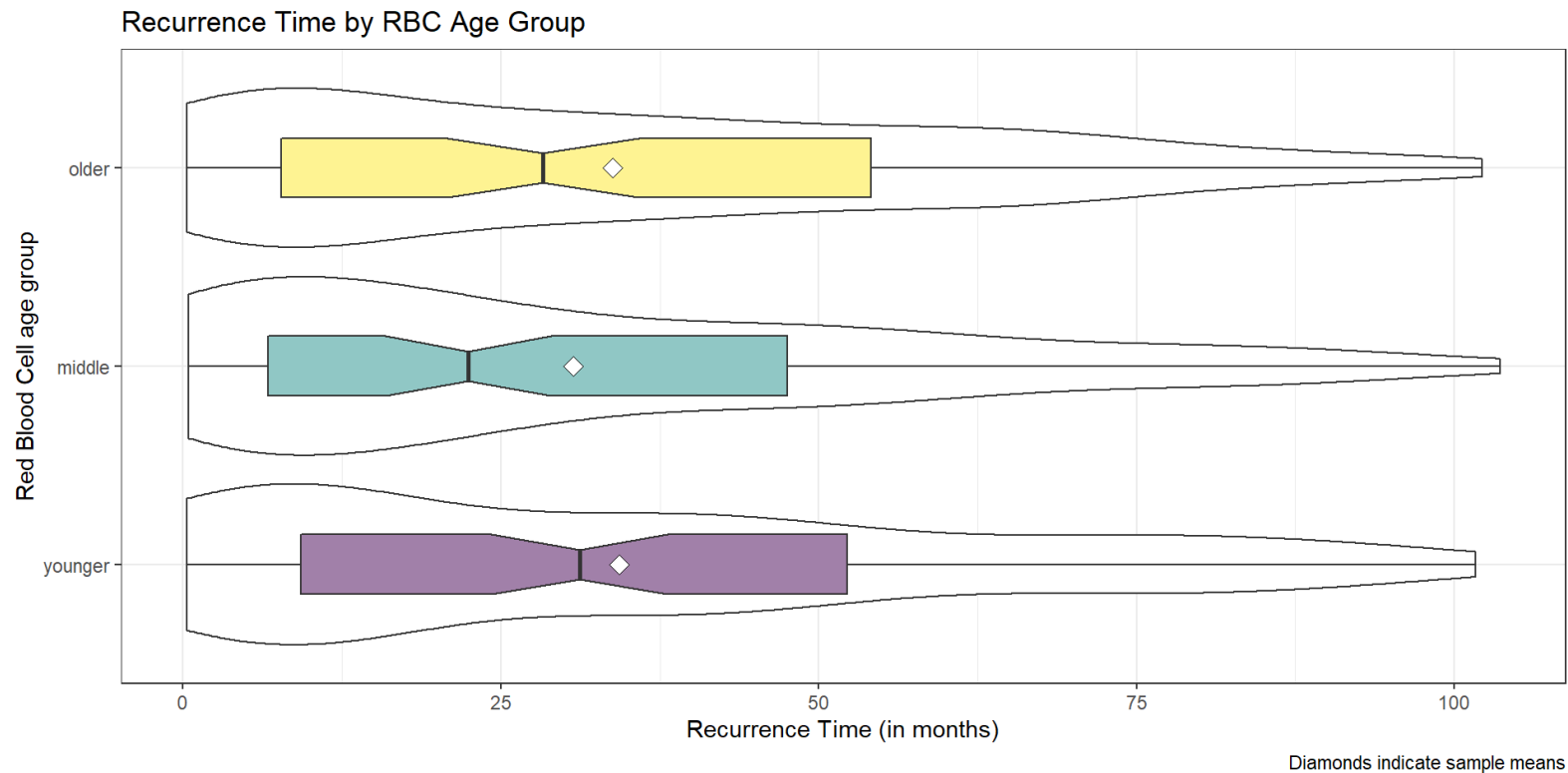
Would a Transformation Help Us?

```
1  mosaic::favstats(~ recur_time, data = bs_cc)
```

```
 min   Q1 median    Q3    max      mean       sd   n missing
0.27  7.6   25.3 52.07  103.6  32.89225 28.47644 289       0
```

Since all `recur_time` values are positive, we might look at:

$log(time)$, or $1/time$, or $\sqrt{time}$, or $time^2$, for example...

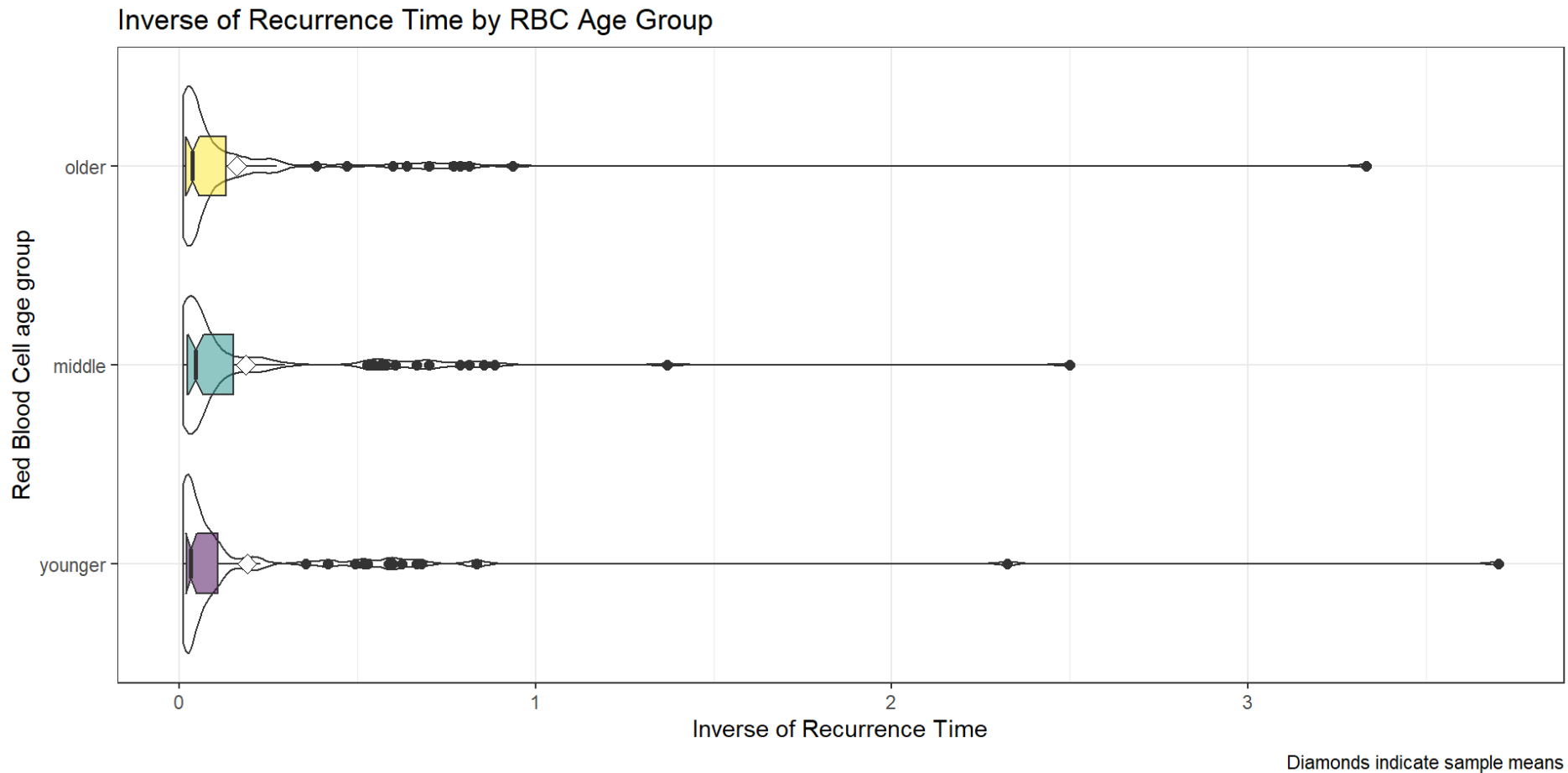What are we hoping these transformations will do?

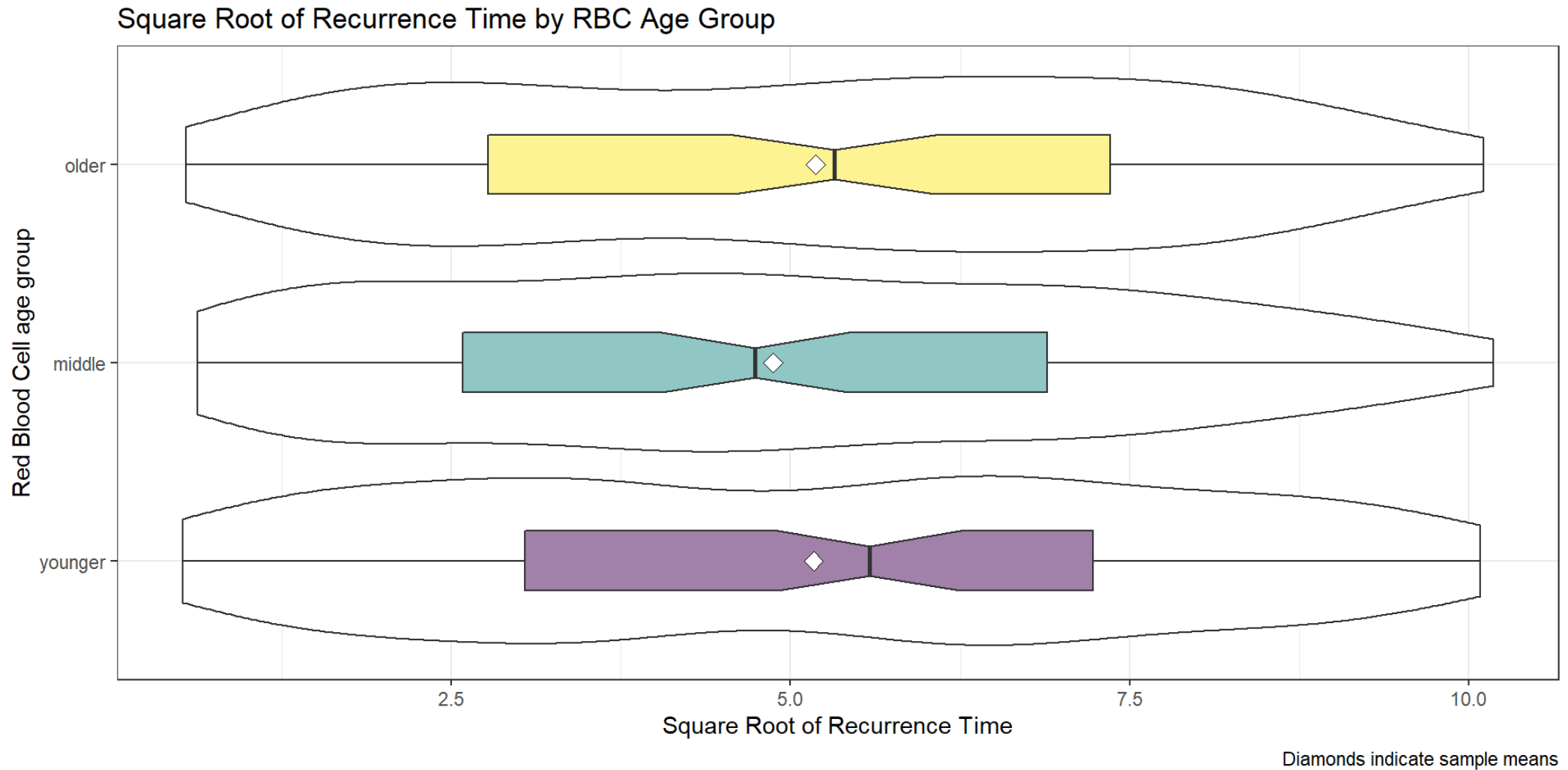# Boxplot 0: `recur_time` by `age_group`

Recurrence Time by RBC Age Group

Diamonds indicate sample means

Boxplot 1: `log(recur_time)` by `age_group`

Natural Log of Recurrence Time by RBC Age Group

Diamonds indicate sample means
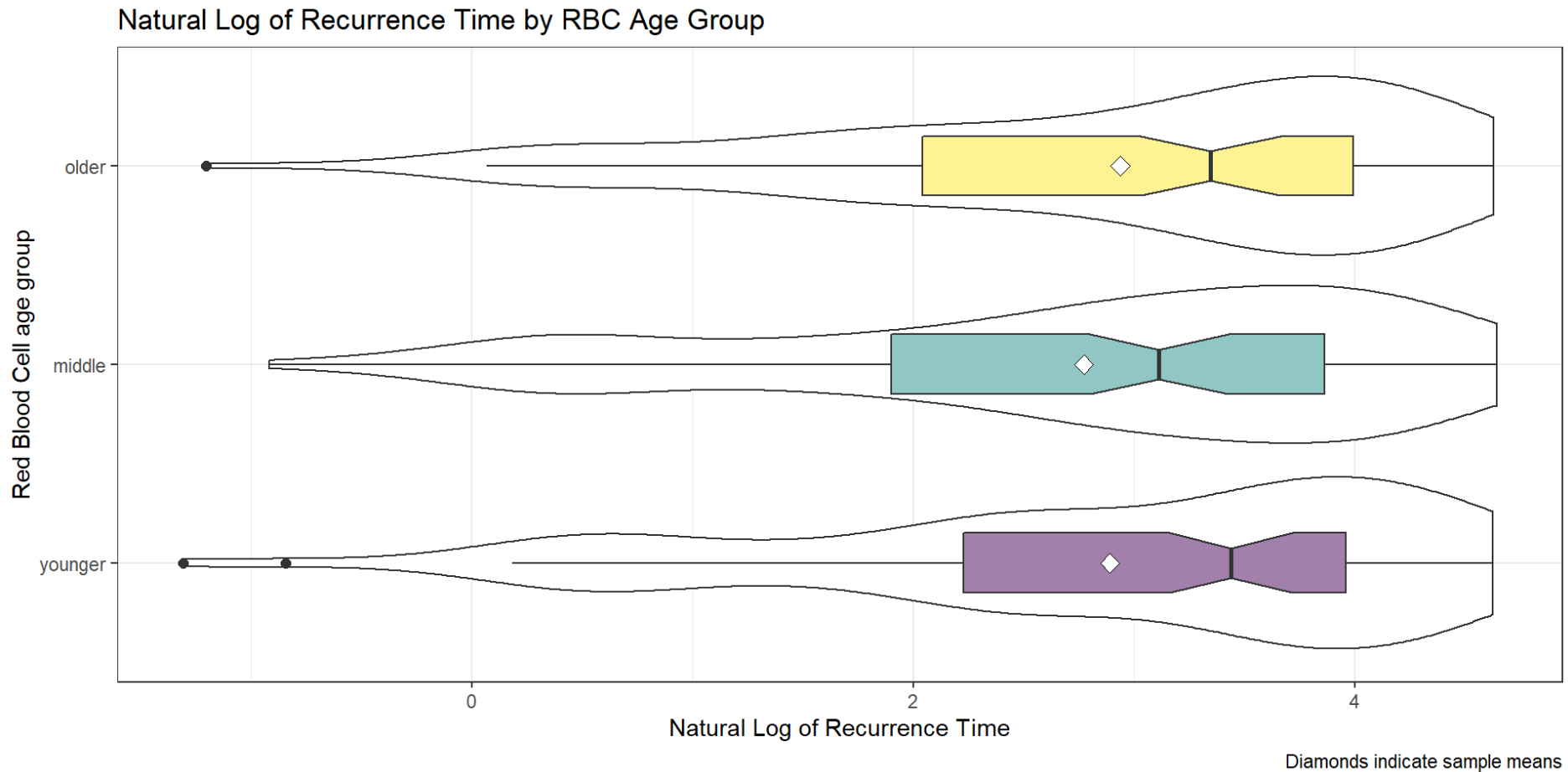
Boxplot 2: `1/(recur_time)` by `age_group`

Inverse of Recurrence Time by RBC Age Group

Diamonds indicate sample means

431

Boxplot 3: $\sqrt{time}$ by `age_group`

Square Root of Recurrence Time by RBC Age Group

Diamonds indicate sample means

Code for Boxplot 3

```
 1  ggplot(data = bs_cc, aes(x = age_group, y = sqrt(recur_time))) +
 2    geom_violin() +
 3    geom_boxplot(aes(fill = age_group), width = 0.3,
 4                 notch = TRUE, outlier.size = 2) +
 5    stat_summary(fun = "mean", geom = "point",
 6                 shape = 23, size = 3, fill = "white") +
 7    guides(fill = "none") +
 8    coord_flip() +
 9    scale_fill_viridis_d(alpha = 0.5) +
10    labs(y = "Square Root of Recurrence Time",
11         x = "Red Blood Cell age group",
12         title = "Square Root of Recurrence Time by RBC Age Group",
13         caption = "Diamonds indicate sample means")
```

431 CASE WESTERN RESERVE UNIVERSITY

Ridgeline Plot for $\sqrt{time}$?

```
1  ggplot(data = bs_cc, aes(x = sqrt(recur_time), y = age_group,
2                         fill = age_group)) +
3    geom_density_ridges(alpha = 0.5) +
4    guides(fill = "none") +
5    labs(title = "Square Root of Time to Recurrence, by RBC Age Group")
```

Square Root of Time to Recurrence, by RBC Age Group
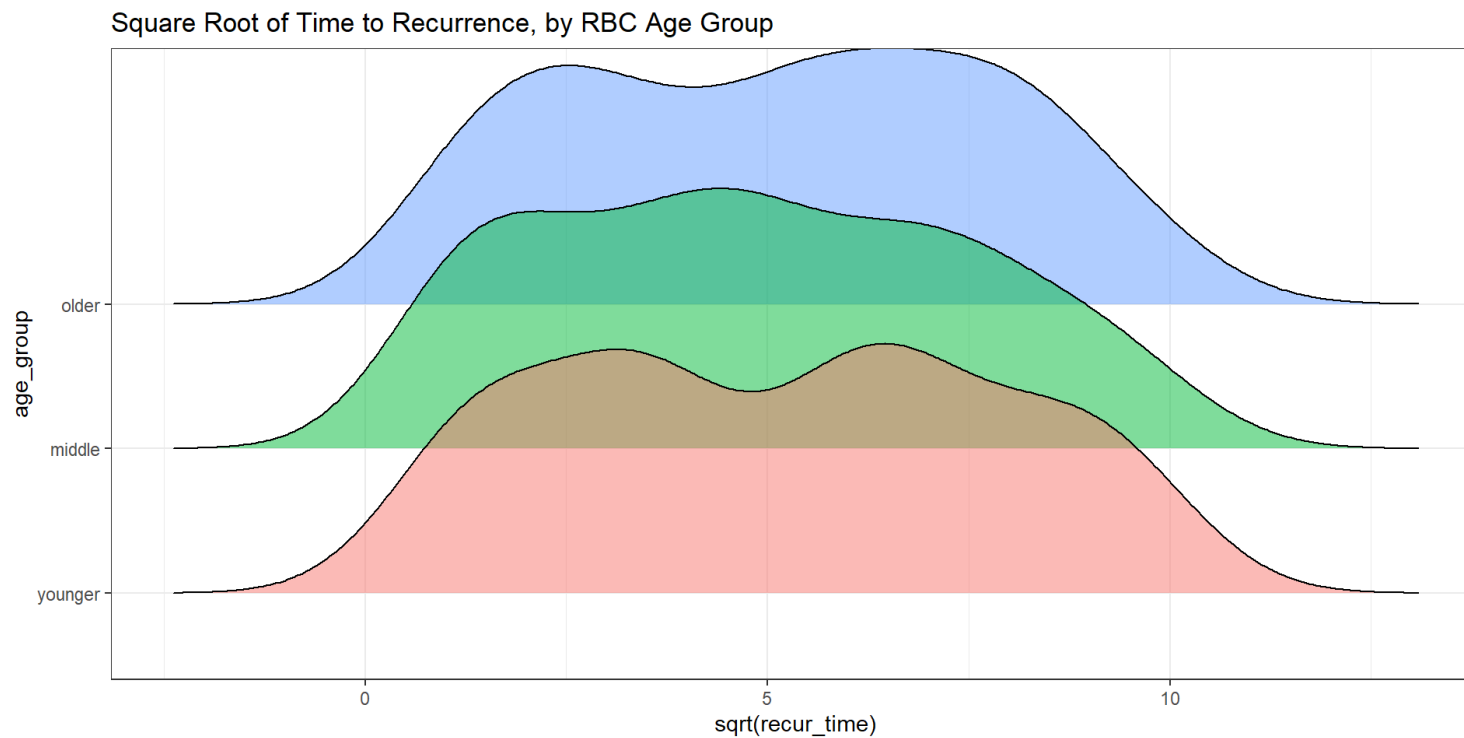
Fit a Model to predict $\sqrt{time}$?

```
1  m2 <- lm(sqrt(recur_time) ~ age_group, data = bs_cc)
2
3  extract_eq(m2, use_coefs = TRUE, wrap = TRUE, coef_digits = 3,
4             terms_per_line = 1, operator_location = "start",
5             font_size = "small")
```

$$\text{sqrt}(\widehat{\text{recur\_time}}) = 5.17$$
$$- 0.299(\text{age\_group}_{\text{middle}})$$
$$+ 0.014(\text{age\_group}_{\text{older}})$$

Predicted Values using m2

$$\text{sqrt}(\widehat{\text{recur\_time}}) = 5.17$$
$$- 0.299(\text{age\_group}_{\text{middle}})$$
$$+ 0.014(\text{age\_group}_{\text{older}})$$

age_group	Est. $\sqrt{time}$	Est. recur_time
Younger	5.17	?
Middle	5.17 - 0.299 = 4.871	?
Older	?	?

Predicted recur_time using m2

$$\mathrm{sqrt}(\widehat{\mathrm{recur\_time}}) = 5.17$$
$$- 0.299(\mathrm{age\_group}_{\mathrm{middle}})$$
$$+ 0.014(\mathrm{age\_group}_{\mathrm{older}})$$

age_group	Est. $\sqrt{time}$	Est. recur_time
Younger	5.17	26.73
Middle	5.17 - 0.299 = 4.871	23.73
Older	5.17 + 0.014 = 5.184	26.87

Tidy model m2

```
1   tidy(m2, conf.int = TRUE, conf.level = 0.90) |>
2     kbl(digits = 2) |> kable_styling(font_size = 28)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	5.17	0.27	18.87	0.00	4.72	5.62
age_groupmiddle	-0.30	0.39	-0.78	0.44	-0.94	0.34
age_groupolder	0.01	0.39	0.03	0.97	-0.63	0.65

glance to summarize m2's fit

```
1   glance(m2) |>
2     select(r.squared, AIC, nobs, df, df.residual) |>
3     kbl(digits = c(4, 1, 0, 0, 0)) |> kable_styling(font_size = 28)
```

r.squared	AIC	nobs	df	df.residual
0.0029	1395.9	289	2	286

431 CASE WESTERN RESERVE UNIVERSITY

Normal Q-Q plot of residuals for m2

```
1  m2_aug <- augment(m2, data = bs_cc)
2
3  ggplot(data = m2_aug, aes(sample = .resid)) +
4    geom_qq() + geom_qq_line(col = "red")
```
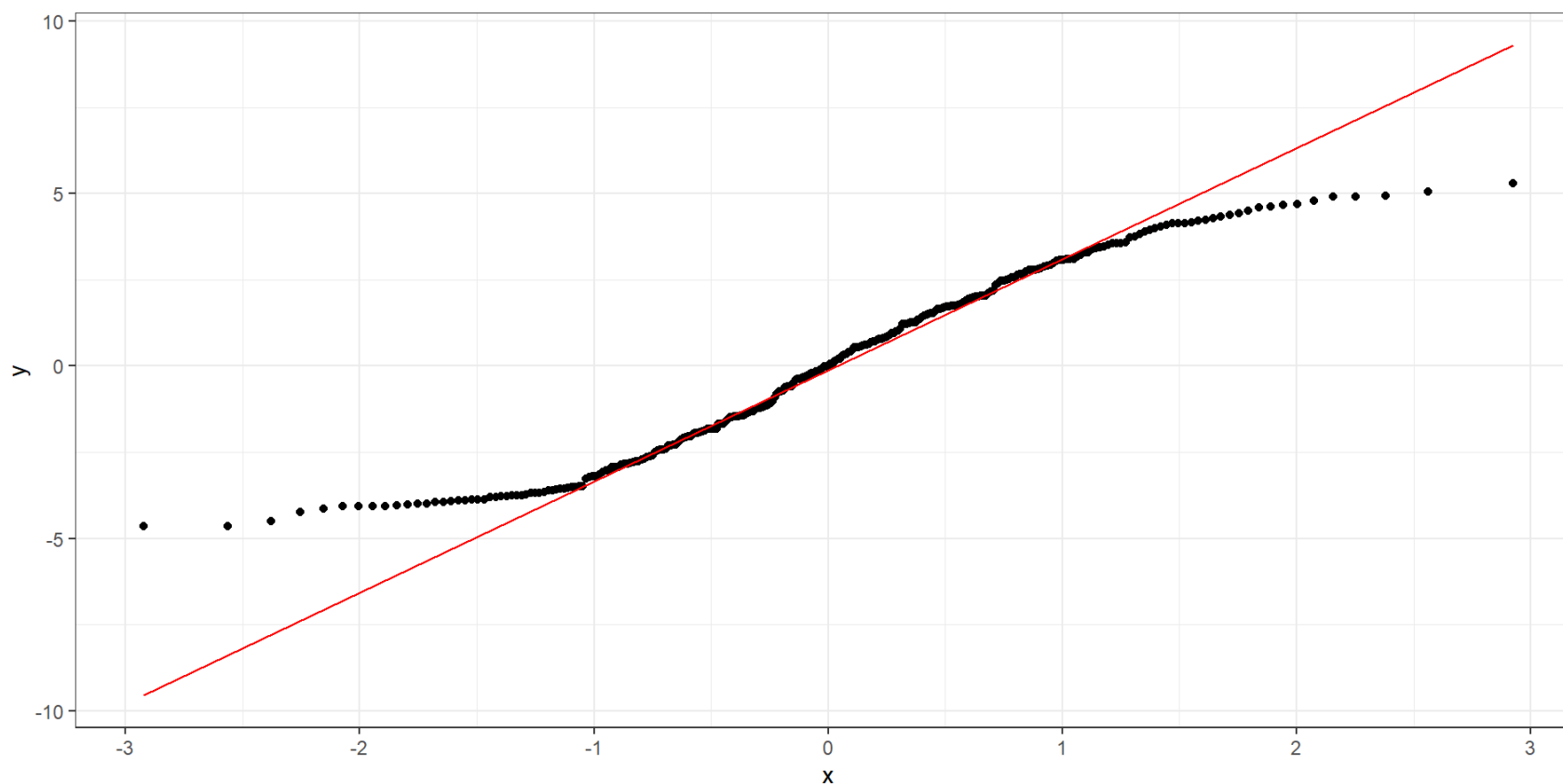
Power Transformations

Tukey's Ladder of Power Transformations

- most useful when the outcome is strictly positive

- most useful when dealing with skew in the outcome

Power	-2	-1	0	0.5	1	2	3
Transformation	$\frac{1}{y^2}$	$\frac{1}{y}$	$log(y)$	$\sqrt{y}$	y	y^2	y^3

- Right Skew usually requires transformations with powers below 1

- Left Skew usually requires powers greater than 1

Consider the **n_income** data in **dm1000**

```
1  dm1000 <- read_rds("c09/data/dm_1000.Rds")
2
3  mosaic::favstats(~ n_income, data = dm1000) |>
4    select(n, missing, min, median, mean, max) |>
5    kbl(digits = 2) |>
6    kable_styling(full_width = FALSE)
```

n	missing	min	median	mean	max
972	28	2279	30586.5	35177.88	129549

Normal Q-Q plot of **n_income**

```
1  dm972 <- dm1000 |> filter(complete.cases(n_income))
2  ggplot(data = dm972, aes(sample = n_income)) +
3    geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1)
```

431 CASE WESTERN RESERVE UNIVERSITY
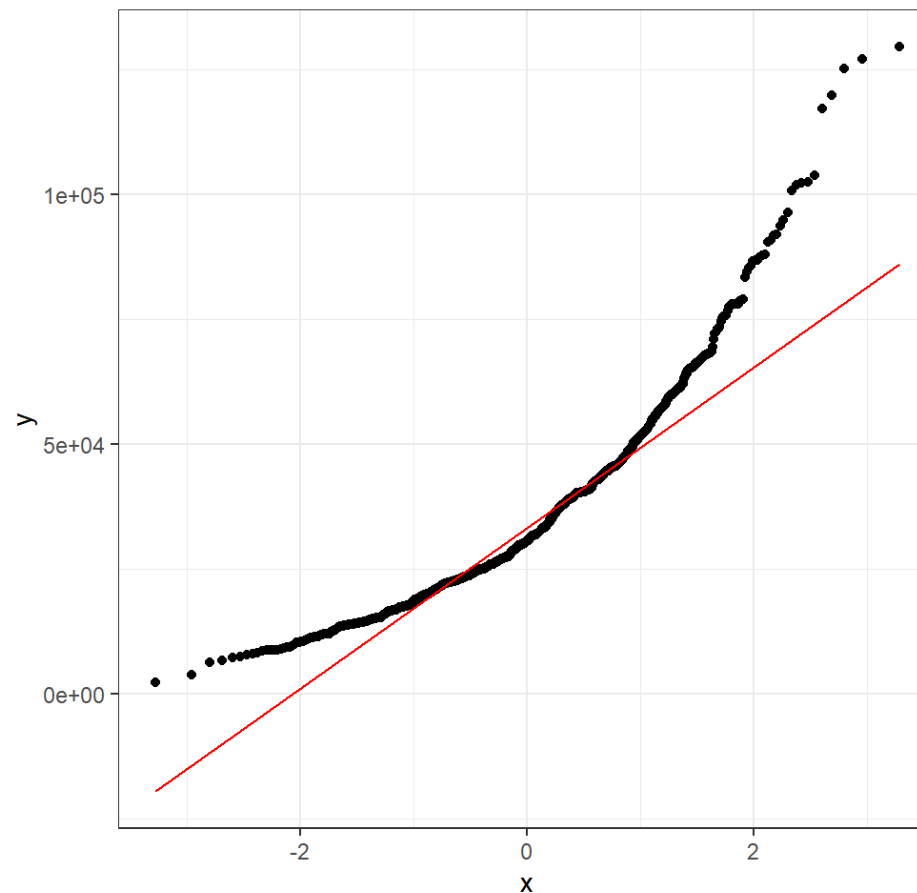
Ladder of `n_income` transformations

```
1  p1 <- ggplot(data = dm972, aes(sample = n_income)) +
2    geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1) +
3    labs(title = "Untransformed", subtitle = "Power = 1", x = "", y = "")
4
5  p2 <- ggplot(data = dm972, aes(sample = n_income^2)) +
6    geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1) +
7    labs(title = "Square", subtitle = "Power = 2", x = "", y = "")
8
9  p3 <- ggplot(data = dm972, aes(sample = n_income^3)) +
10   geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1) +
11   labs(title = "Cube", subtitle = "Power = 3", x = "", y = "")
12
13 p4 <- ggplot(data = dm972, aes(sample = sqrt(n_income))) +
14   geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1) +
15   labs(title = "Square Root", subtitle = "Power = 1/2", x = "", y = "")
16
17 p5 <- ggplot(data = dm972, aes(sample = log(n_income))) +
18   geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1) +
19   labs(title = "Logarithm", subtitle = "Power = 0", x = "", y = "")
```
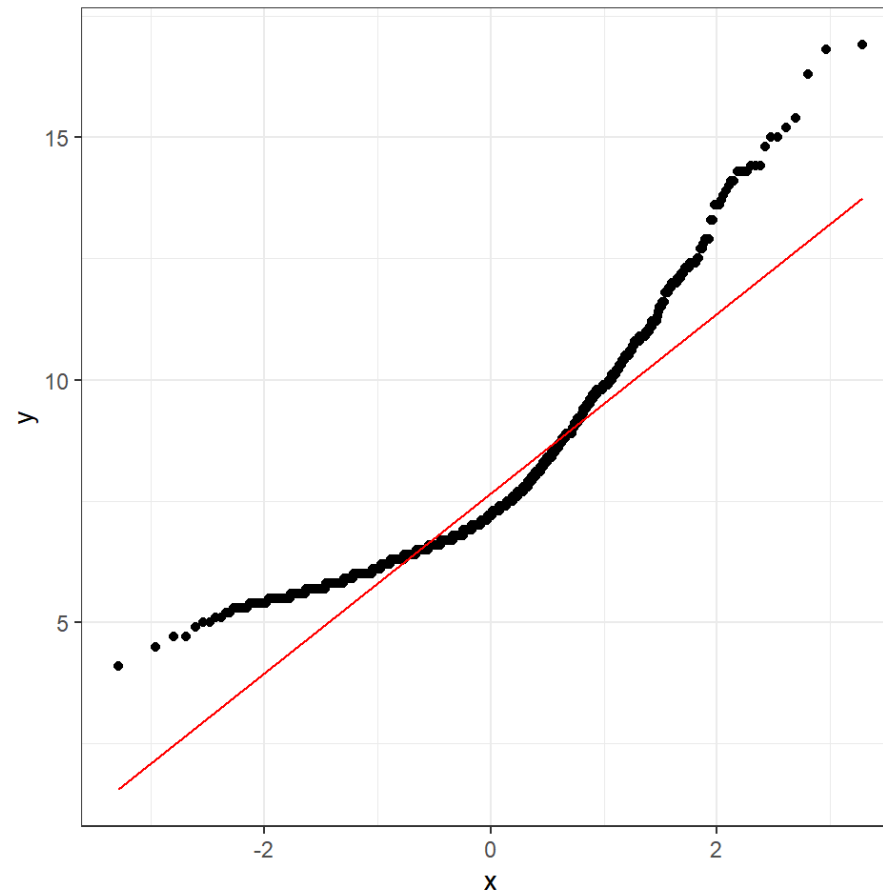
Ladder of `n_income` transformations
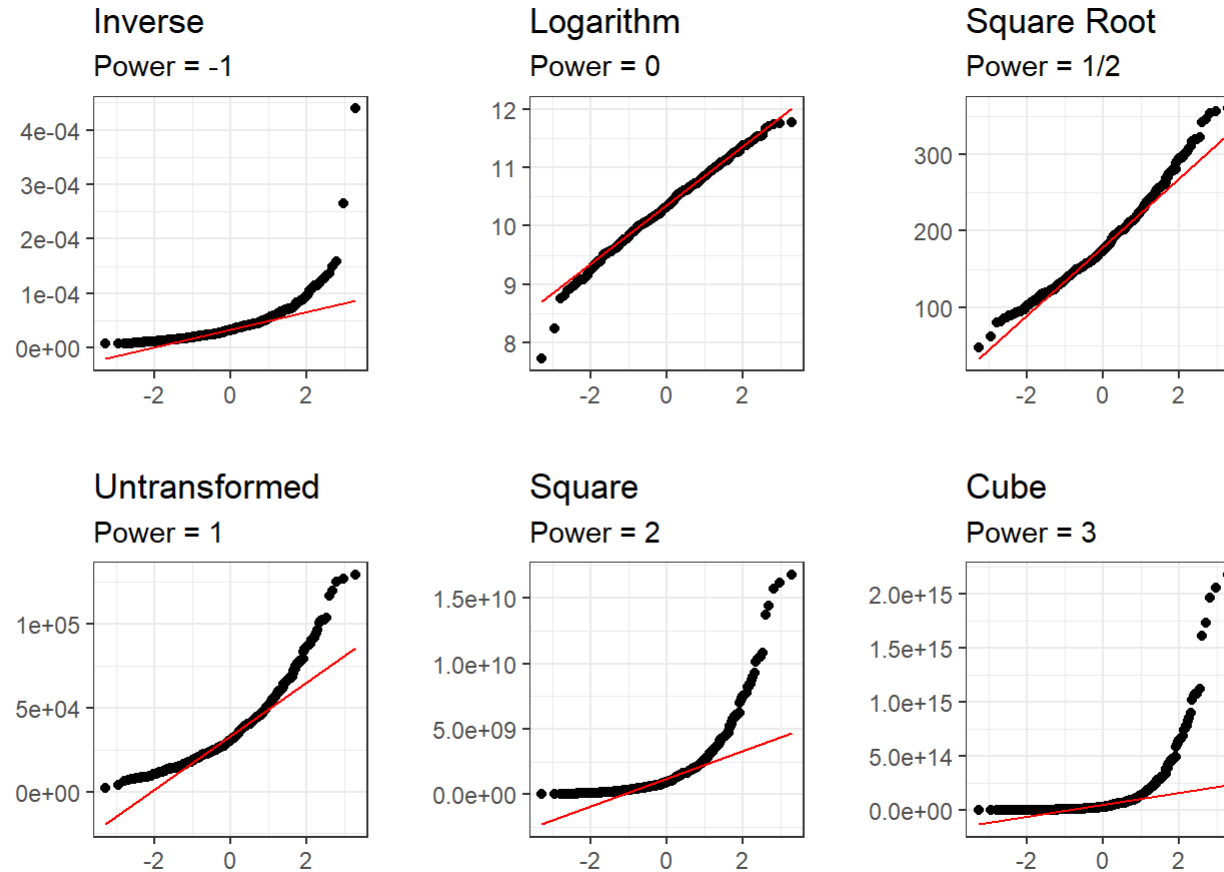
Hemoglobin A1c data in dm1000

```
1  dm985 <- dm1000 |> filter(complete.cases(a1c))
2  ggplot(data = dm985, aes(sample = a1c)) +
3    geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1)
```

431

Ladder of A1c transformations

```
 1  p1 <- ggplot(data = dm985, aes(sample = a1c)) +
 2    geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1) +
 3    labs(title = "Untransformed", subtitle = "Power = 1", x = "", y = "")
 4
 5  p2 <- ggplot(data = dm985, aes(sample = a1c^2)) +
 6    geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1) +
 7    labs(title = "Square", subtitle = "Power = 2", x = "", y = "")
 8
 9  p3 <- ggplot(data = dm985, aes(sample = a1c^3)) +
10    geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1) +
11    labs(title = "Cube", subtitle = "Power = 3", x = "", y = "")
12
13  p4 <- ggplot(data = dm985, aes(sample = sqrt(a1c))) +
14    geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1) +
15    labs(title = "Square Root", subtitle = "Power = 1/2", x = "", y = "")
16
17  p5 <- ggplot(data = dm985, aes(sample = log(a1c))) +
18    geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1) +
19    labs(title = "Logarithm", subtitle = "Power = 0", x = "", y = "")
```

Ladder of A1c transformations
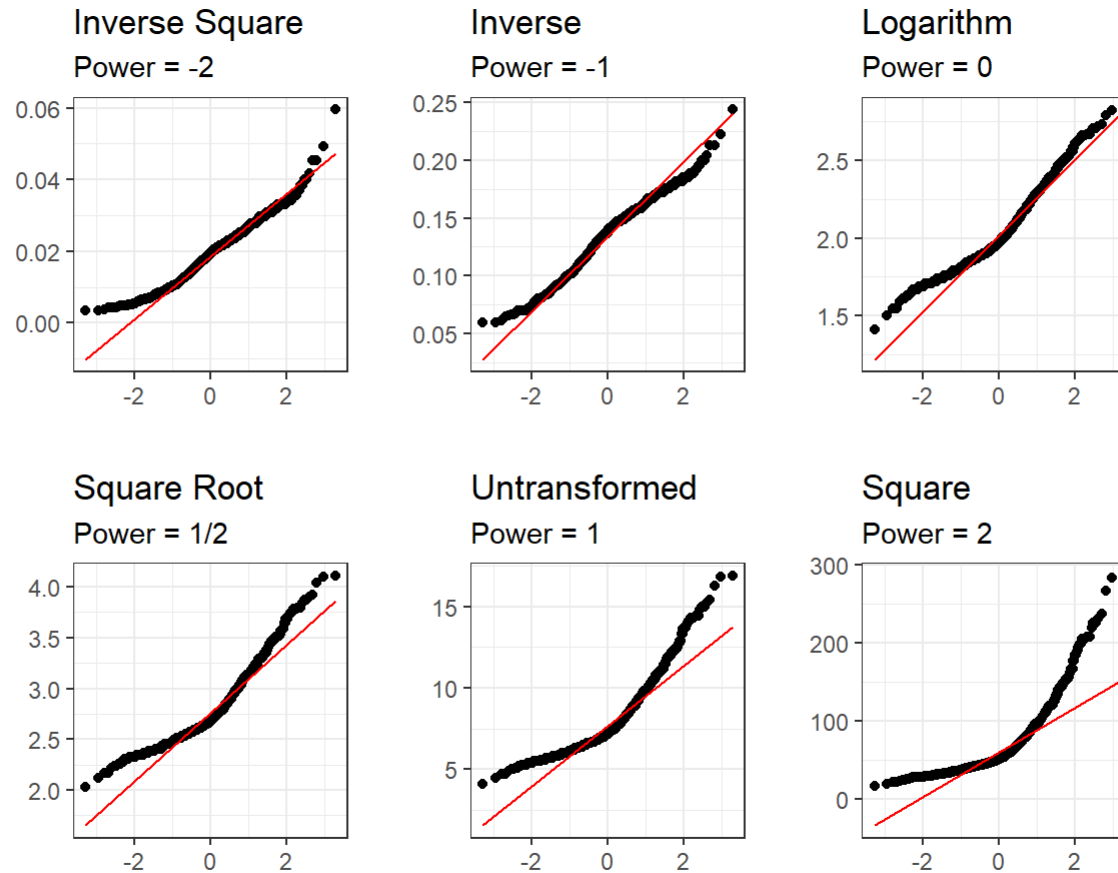
An Example to Work through on your own

Predict time with `units`

Some data prep required:

- `units` is actually a count.

- Use all 291 observations with `recur_time` and `units`.

```
1  bs_dat2 <- bs_dat |>
2    filter(complete.cases(recur_time, units))
3
4  bs_dat2 |> tabyl(units)
```

```
 units    n    percent
     1   67  0.2302405
     2  174  0.5979381
     3   19  0.0652921
     4   31  0.1065292
```

Scatterplot of `recur_time` vs. `age_group`

```r
1  ggplot(bs_dat2, aes(x = age_group, y = recur_time)) +
2    geom_point() + geom_smooth(method = "lm", se = FALSE)
```

Comparison Boxplot

```
1  ggplot(data = bs_dat2, aes(x = factor(units), y = recur_time)) +
2    geom_violin() +
3    geom_boxplot(aes(fill = factor(units)), width = 0.3,
4                 outlier.size = 2) +
5    stat_summary(fun = "mean", geom = "point",
6                 shape = 23, size = 3, fill = "white") +
7    guides(fill = "none") +
8    coord_flip() +
9    scale_fill_viridis_d(alpha = 0.5) +
10   labs(y = "Recurrence Time (in months)",
11        x = "Number of Units Received",
12        title = "Recurrence Time by Units",
13        caption = "Diamonds indicate sample means")
```

Comparison Boxplot

Recurrence Time by Units
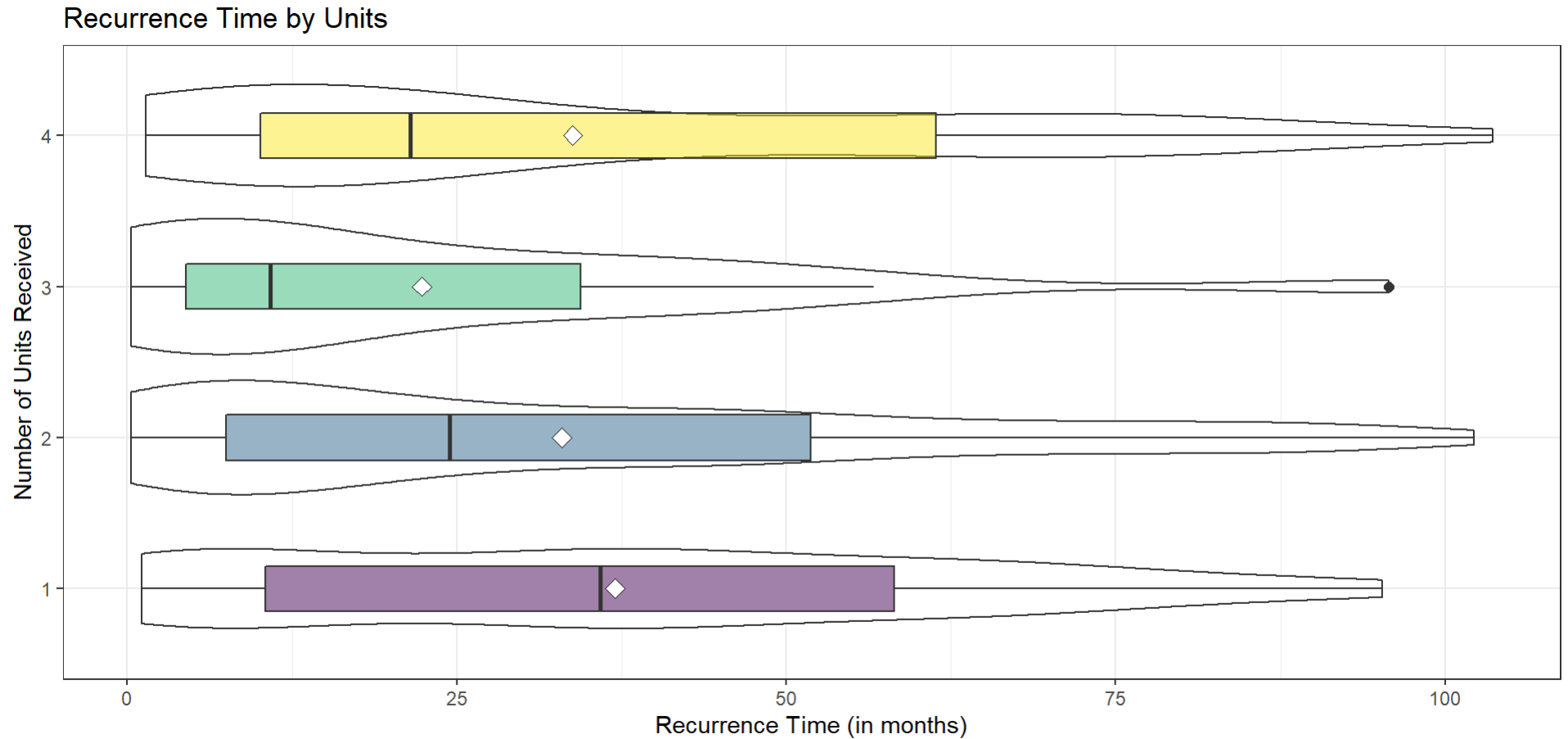
Model Time using Units

```
1  m3 <- lm(recur_time ~ units, data = bs_dat2)
2
3  extract_eq(m3, use_coefs = TRUE, coef_digits = 2)
```

$$\widehat{\text{recur\_time}} = 37.47 - 2.04(\text{units})$$

```
1  tidy(m3, conf.int = TRUE, conf.level = 0.90)
```

```
# A tibble: 2 × 7
  term        estimate std.error statistic  p.value conf.low conf.high
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
1 (Intercept)     37.5      4.41      8.50 1.06e-15     30.2      44.8
2 units           -2.04     1.99     -1.03 3.06e- 1     -5.32      1.24
```

431 CASE WESTERN RESERVE UNIVERSITY

Model Square Root of Time using Units

```
1  m4 <- lm(sqrt(recur_time) ~ units, data = bs_dat2)
2
3  extract_eq(m4, use_coefs = TRUE, coef_digits = 2)
```

$$\mathrm{sqrt}(\widehat{\mathrm{recur\_time}}) = 5.54 - 0.21(\mathrm{units})$$

```
1  tidy(m4, conf.int = TRUE, conf.level = 0.90)
```

```
# A tibble: 2 × 7
  term          estimate std.error statistic  p.value conf.low conf.high
  <chr>            <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
1 (Intercept)      5.54     0.413     13.4  3.33e-32     4.85      6.22
2 units           -0.211    0.186     -1.13 2.59e- 1    -0.518     0.0967
```

431 CASE WESTERN RESERVE UNIVERSITY

Normal Q-Q plots of Residuals

```
1  m3_aug <- augment(m3, data = bs_dat2)
2  m4_aug <- augment(m4, data = bs_dat2)
3
4  p1 <- ggplot(m3_aug, aes(sample = .resid)) +
5    geom_qq() + geom_qq_line(col = "red") +
6    theme(aspect.ratio = 1) +
7    labs(title = "Model m3 Residuals", x = "", y = "")
8
9  p2 <- ggplot(m4_aug, aes(sample = .resid)) +
10   geom_qq() + geom_qq_line(col = "red") +
11   theme(aspect.ratio = 1) +
12   labs(title = "Model m4 Residuals", x = "", y = "")
13
14 p1 + p2
```

Normal Q-Q plots of Residuals
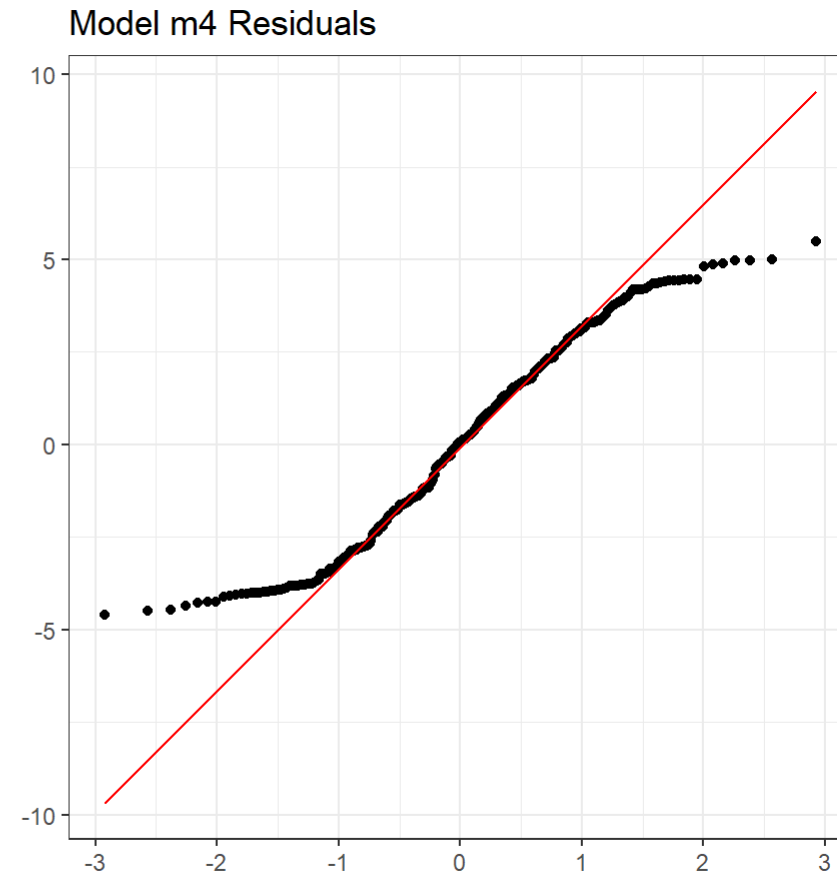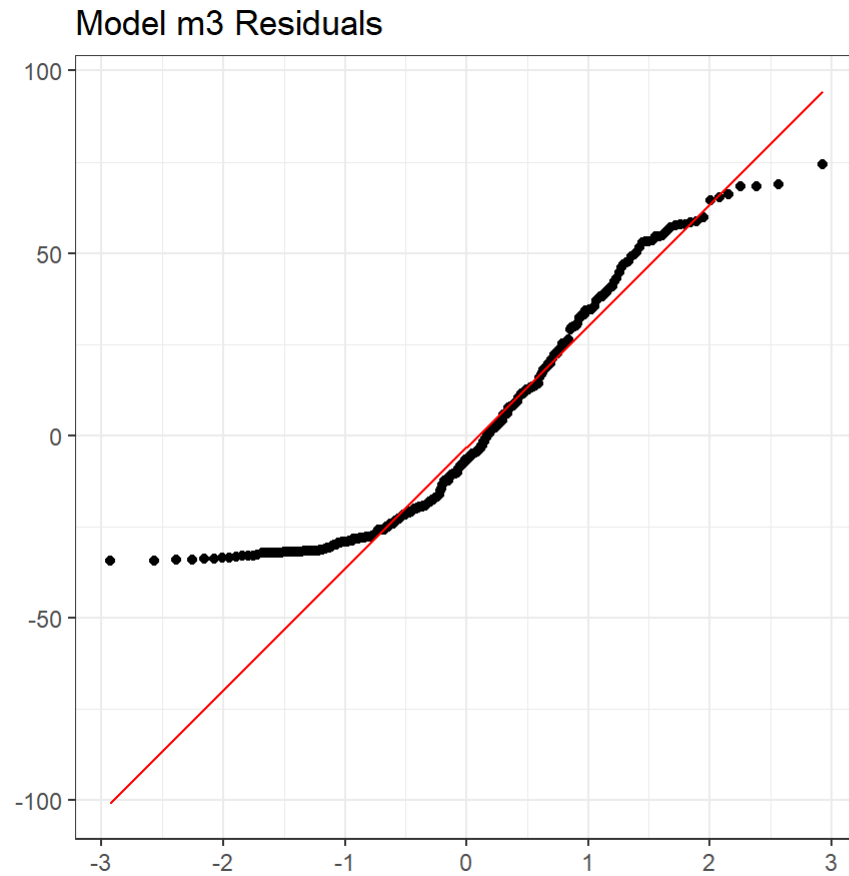
Compare fits of m1 and m3?

```
1  glance(m1) |> select(r.squared, AIC, df, df.residual, nobs)
```

```
# A tibble: 1 × 5
  r.squared    AIC     df df.residual   nobs
      <dbl>  <dbl>  <dbl>       <int>  <int>
1   0.00318  2762.      2         286    289
```

```
1  glance(m3) |> select(r.squared, AIC, df, df.residual, nobs)
```

```
# A tibble: 1 × 5
  r.squared    AIC     df df.residual   nobs
      <dbl>  <dbl>  <dbl>       <int>  <int>
1   0.00362  2785.      1         289    291
```

Are these two models actually predicting the same outcome?

- for the same subjects?

Session Information

```
1 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
```