431 Class 09

Thomas E. Love, Ph.D.

2022-09-27

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

Today’s Setup

```
knitr::opts_chunk$set(comment=NA)
library(broom)               ## tidy up model output
library(equatiomatic)        ## pull equations from regressions
library(ggrepel)             ## build useful labels in ggplot2
library(ggridges)            ## help with ridgeline plots
library(glue)                ## work with strings
library(kableExtra)          ## tidy up tables of output
library(janitor)
library(naniar)
library(simputation)
library(patchwork)
library(tidyverse)

theme_set(theme_bw())
```

Today’s Data

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

```r
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?

```
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?

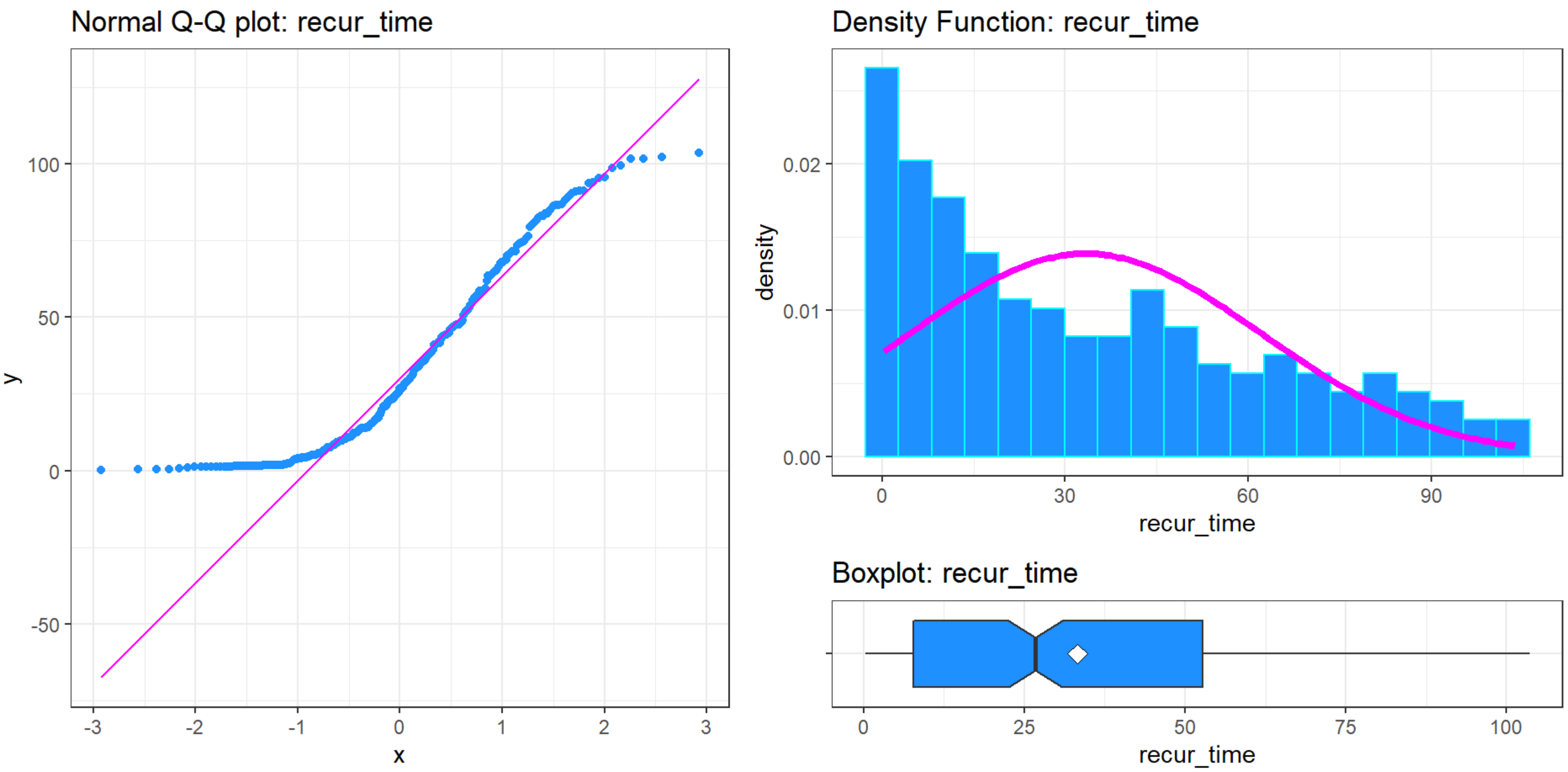
```
miss_var_summary(bs_dat)
```

```
# 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    
```

Outcome is time to recurrence

```r
p1 <- ggplot(bs_dat, aes(sample = recur_time)) +
  geom_qq(col = "dodgerblue") +
  geom_qq_line(col = "magenta") +
  theme(aspect.ratio = 1) +
  labs(title = "Normal Q-Q plot: recur_time")

p2 <- ggplot(bs_dat, aes(x = recur_time)) +
  geom_histogram(aes(y = stat(density)),
                 bins = 20, fill = "dodgerblue", col = "cyan") +
  stat_function(fun = dnorm,
                args = list(mean = mean(bs_dat$recur_time, na.rm = TRUE),
                            sd = sd(bs_dat$recur_time, na.rm = TRUE)),
                col = "magenta", lwd = 1.5) +
  labs(title = "Density Function: recur_time")

p3 <- ggplot(bs_dat, aes(x = recur_time, y = "")) +
  geom_boxplot(fill = "dodgerblue", notch = TRUE,
               outlier.color = "dodgerblue") +
  stat_summary(fun = "mean", geom = "point",
```

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.

```
bs_cc <- bs_dat |> filter(complete.cases(age_group, recur_time, units))

mosaic::favstats(recur_time ~ age_group, data = bs_cc) |>
  kbl(digits = 2) |>
  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`

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

Visualizing Strategies

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

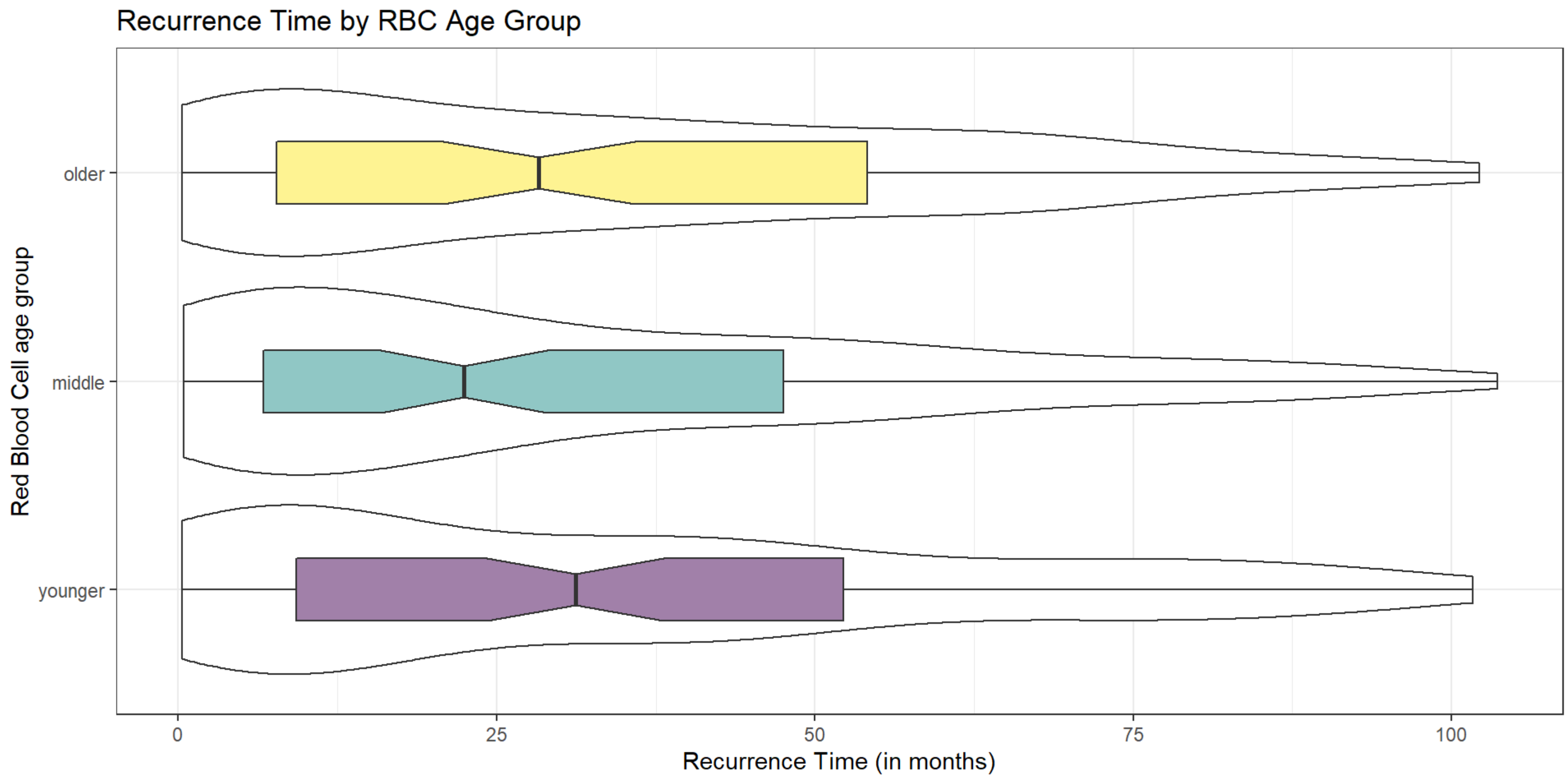
- Comparison Boxplot
- Faceted Histograms
- Overlapping Density Plot
- Ridgeline Plot

So let's walk through each of these.

Comparison Boxplot

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

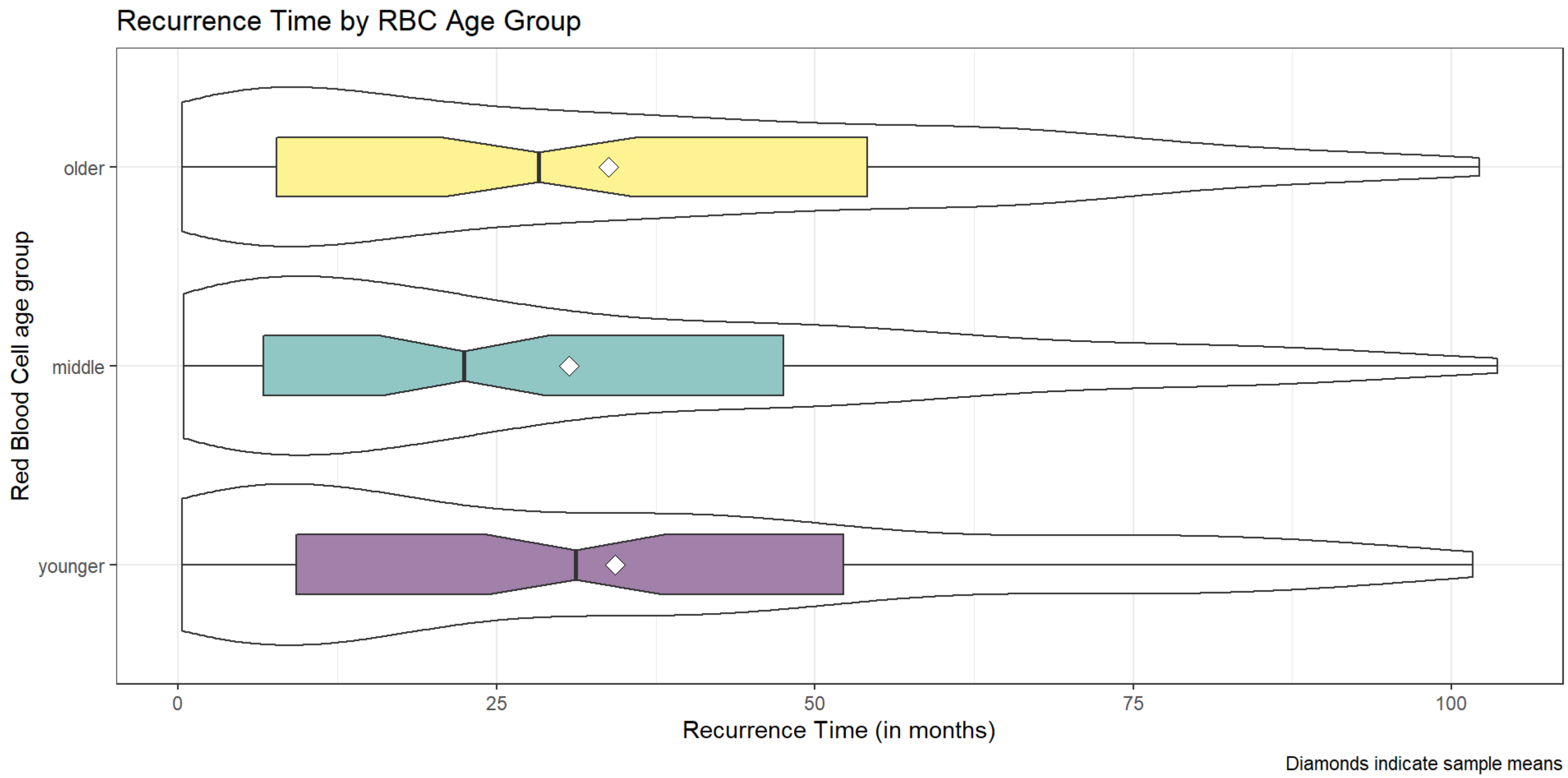
Comparison Boxplot



Add MEANS to Comparison Boxplot

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

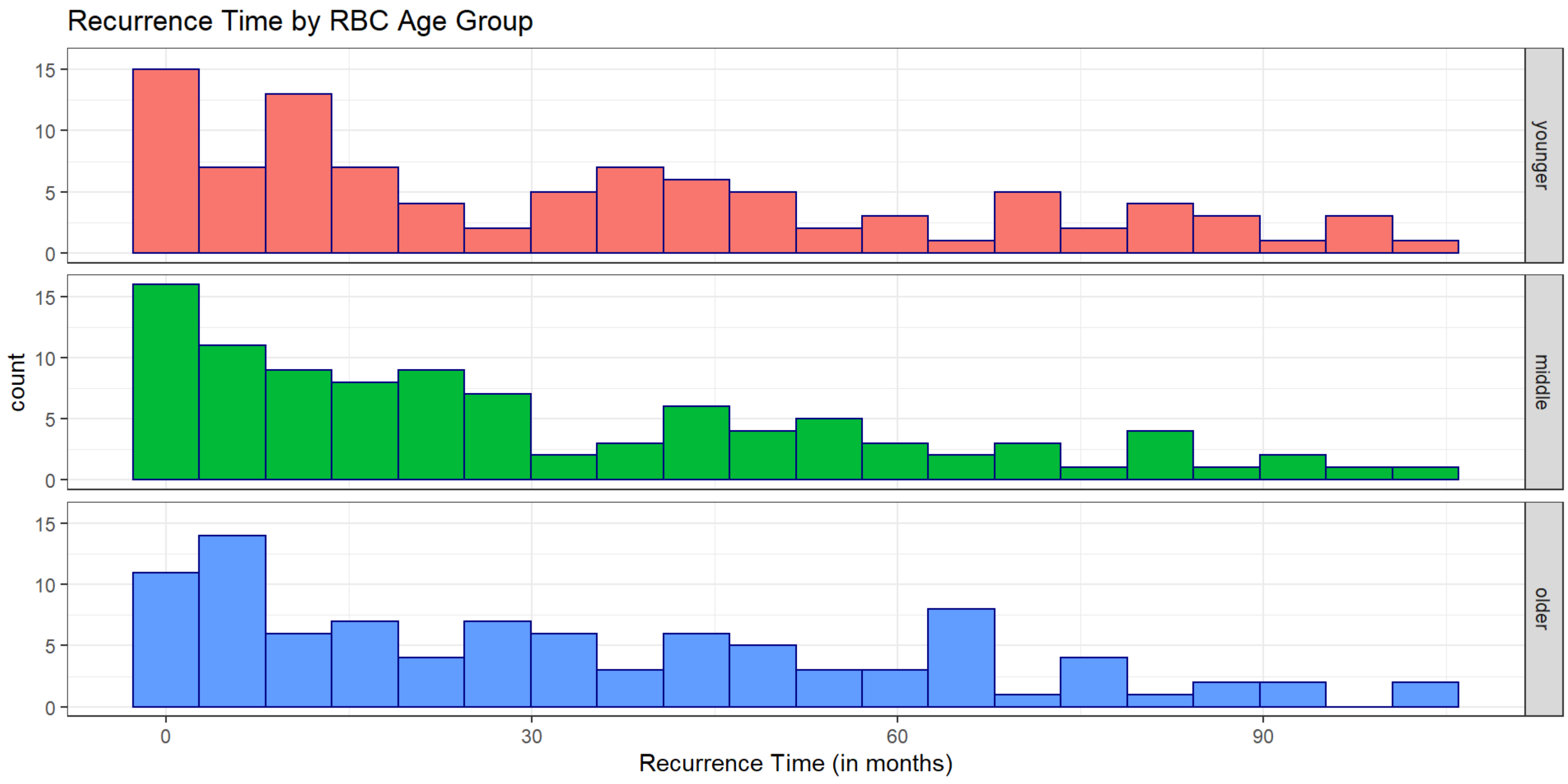
Add MEANS to Comparison Boxplot



Faceted Histograms

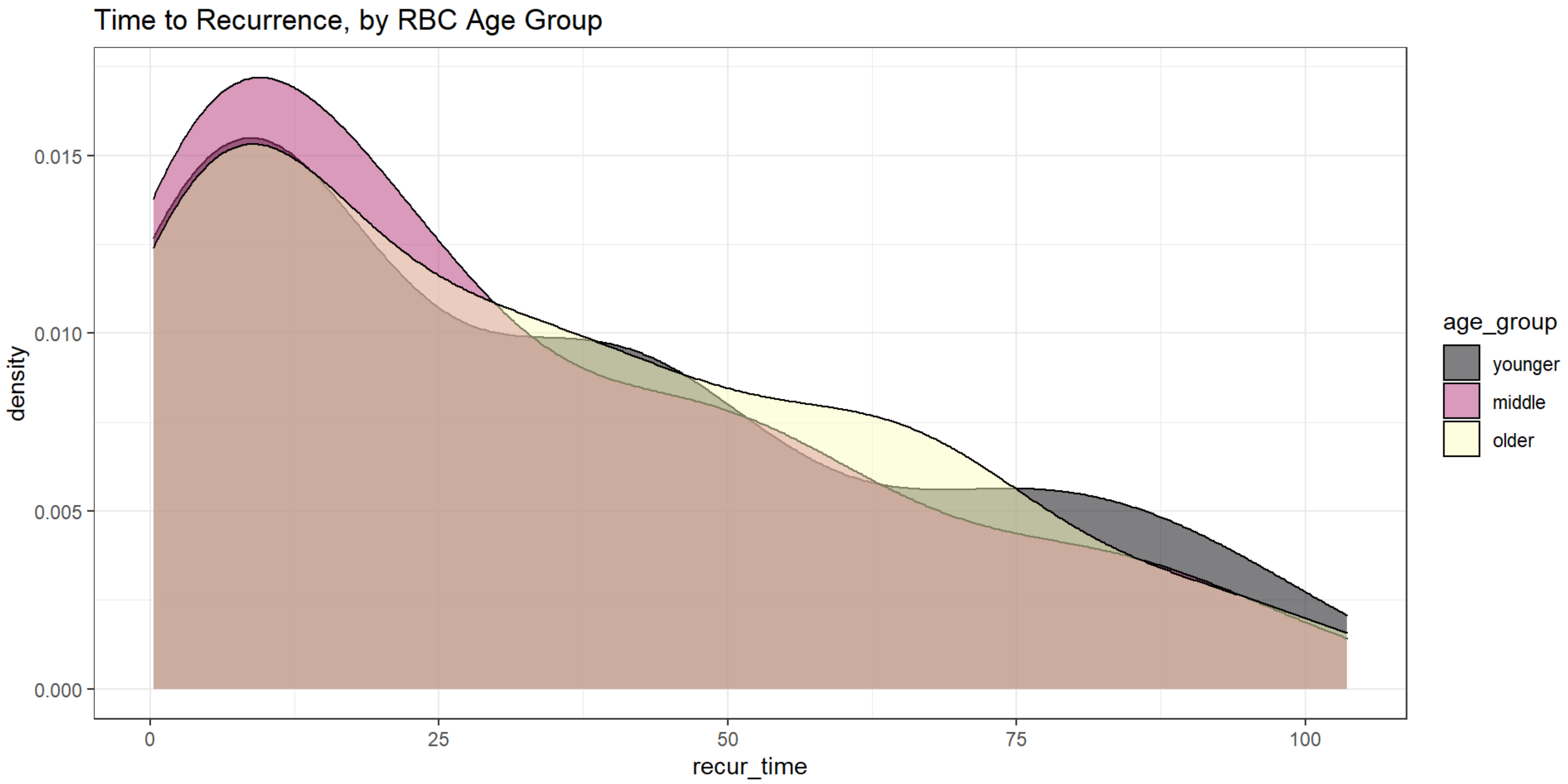
```r
ggplot(data = bs_cc, aes(x = recur_time, fill = age_group)) +
  geom_histogram(bins = 20, col = "navy") +
  guides(fill = "none") +
  facet_grid(age_group ~ .) +
  labs(x = "Recurrence Time (in months)",
       title = "Recurrence Time by RBC Age Group")
```

Faceted Histograms



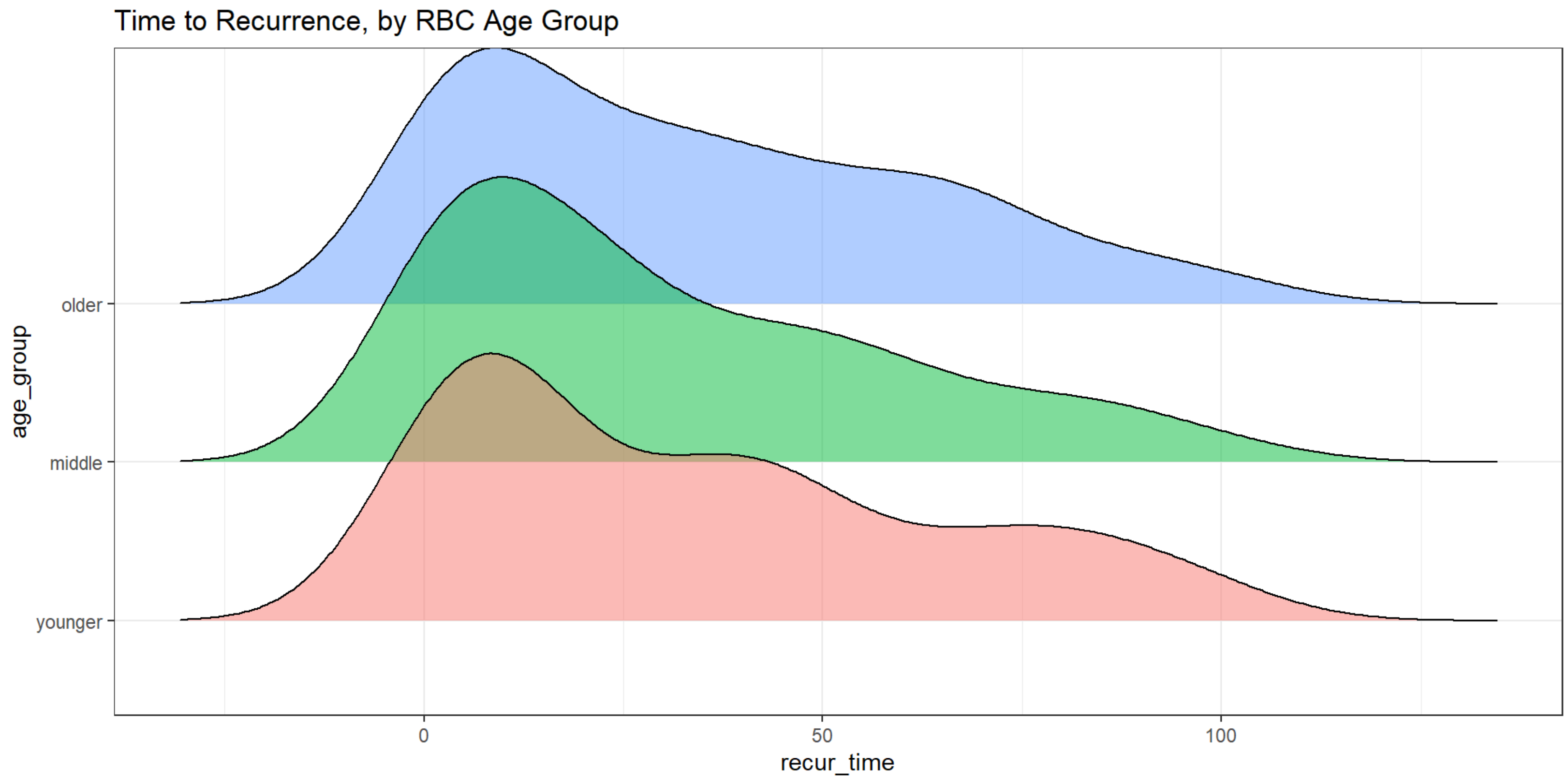
Comparing Densities

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



Using a Ridgeline Plot

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



Complete Cases: Model Time using Age

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

m1
```

```

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

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

Extract Equation with `equatiomatic`

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

$$
\begin{aligned}
\widehat{\operatorname{recur\_time}} &= 34.29 \\
&\quad - 3.61(\operatorname{age\_group}_{\operatorname{middle}}) \\
&\quad - 0.52(\operatorname{age\_group}_{\operatorname{older}})
\end{aligned}
$$

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

`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	

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

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

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

`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

Sample Means from bs_cc

```
mosaic::favstats(recur_time ~ age_group, data = bs_cc) |>
  select(age_group, mean) |>
  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

```
tidy(m1, conf.int = TRUE, conf.level = 0.90) |>
  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?

`glance` to summarize `m1`’s fit

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

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

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?

```
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?

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.

```r
bs_imp <- bs_dat |>
  impute_pmm(recur_time ~ age_group + units) |>
  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

```
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        
```

```
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`

```r
m1_imp <- lm(recur_time ~ age_group, data = bs_imp)

extract_eq(m1_imp, use_coefs = TRUE, wrap = TRUE, coef_digits = 2,
           terms_per_line = 1, operator_location = "start",
           font_size = "small")
```

$$
\begin{aligned}
\widehat{\operatorname{recur\_time}} &= 34.85 \\
&\quad - 4.18(\operatorname{age\_group}_{\operatorname{middle}}) \\
&\quad - 0.46(\operatorname{age\_group}_{\operatorname{older}})
\end{aligned}
$$

Compare Tidied Coefficients

```
tidy(m1, conf.int = TRUE, conf.level = 0.90) |>
  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

```
tidy(m1_imp, conf.int = TRUE, conf.level = 0.90) |>
  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

Compare Summaries with `glance`

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

```r
glance(m1_imp) |>
  select(r.squared, AIC, nobs, df, df.residual) |>
  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

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.

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.

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…

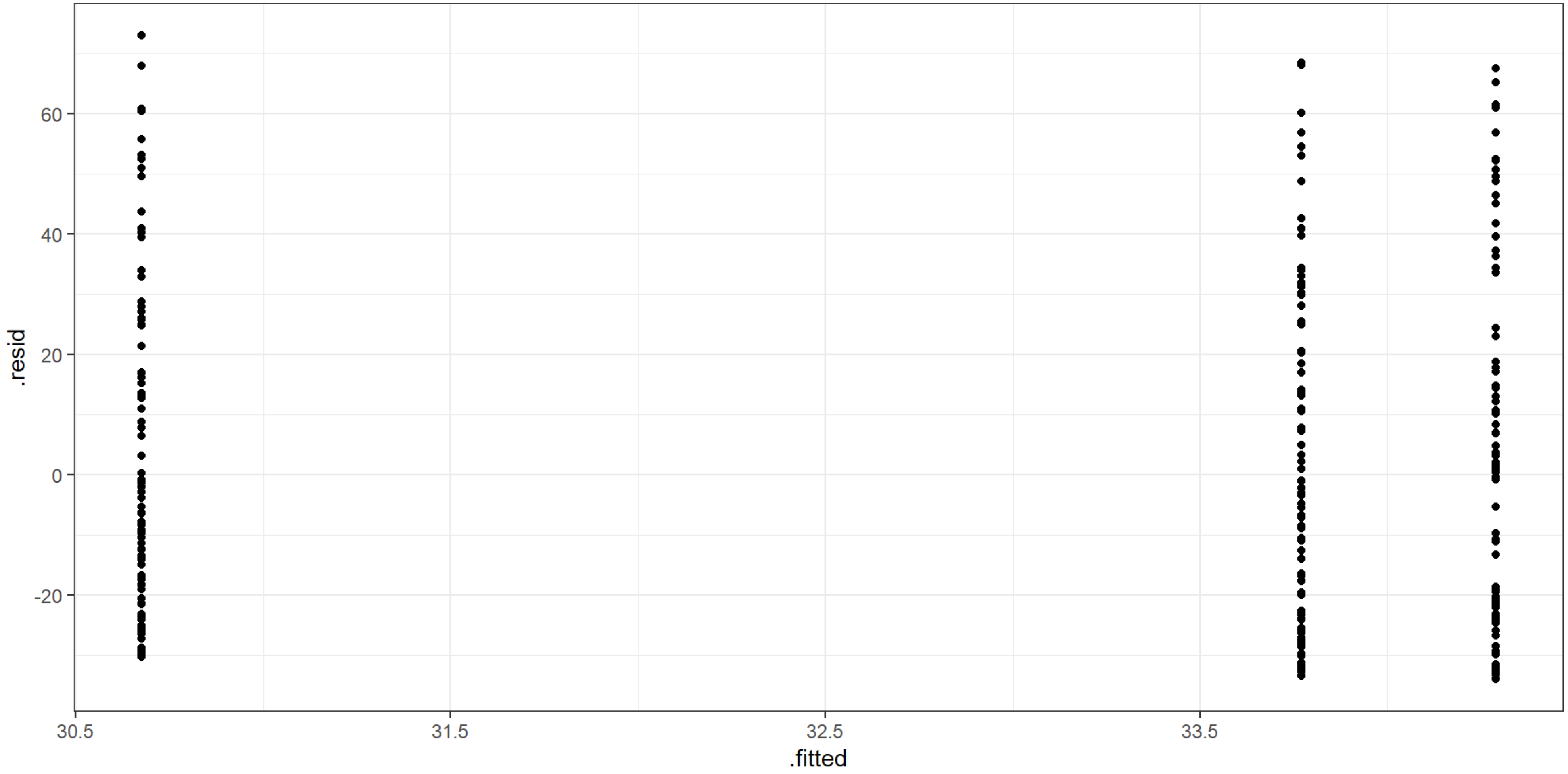
Save residuals and fitted values for m1

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

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   13.9     33.8 -19.9  0.0105   28.6 1.74e-3
```

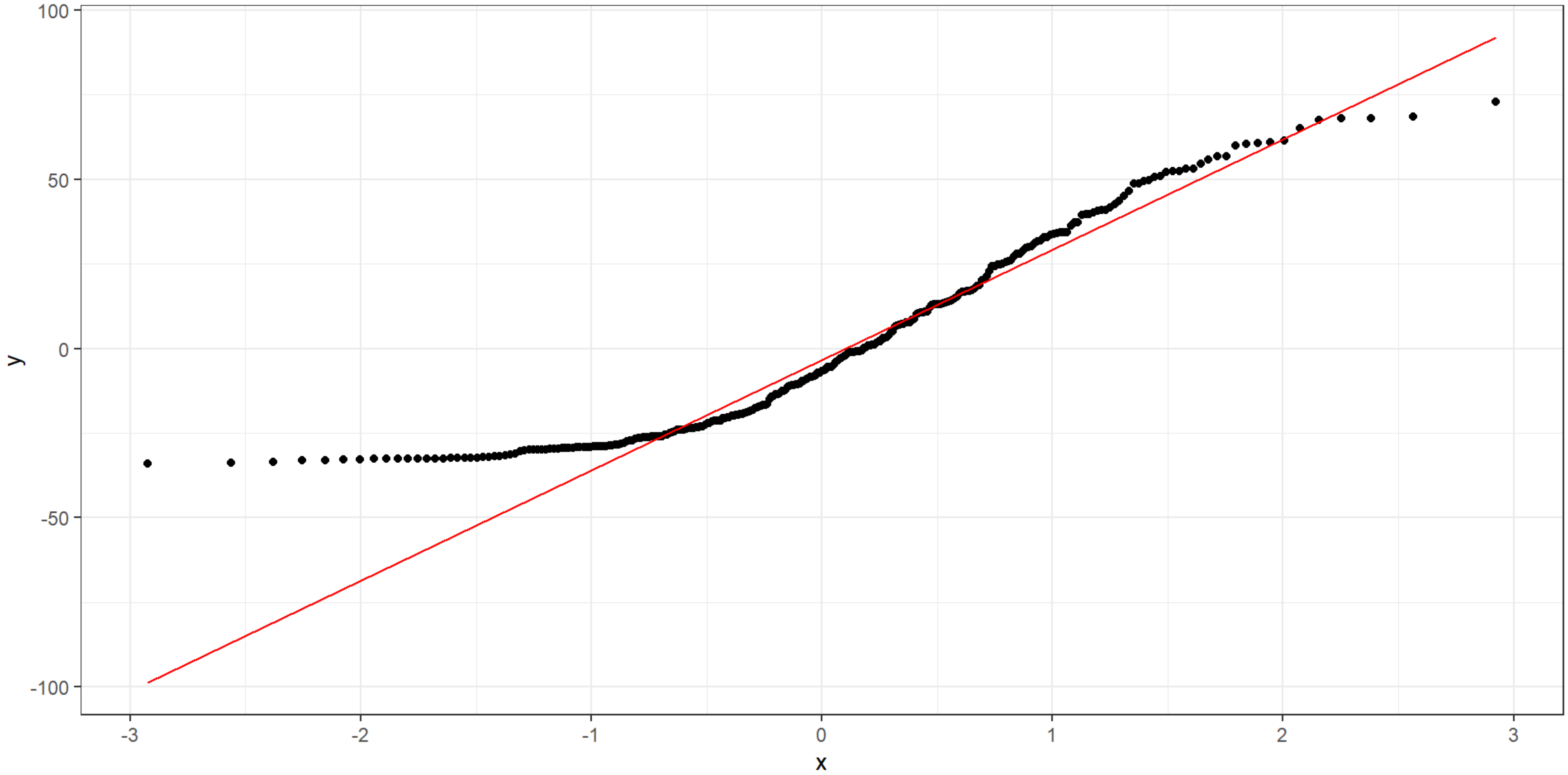
m1 Residuals vs. Fitted Values

```
ggplot(data = m1_aug, aes(x = .fitted, y = .resid)) +
  geom_point()
```



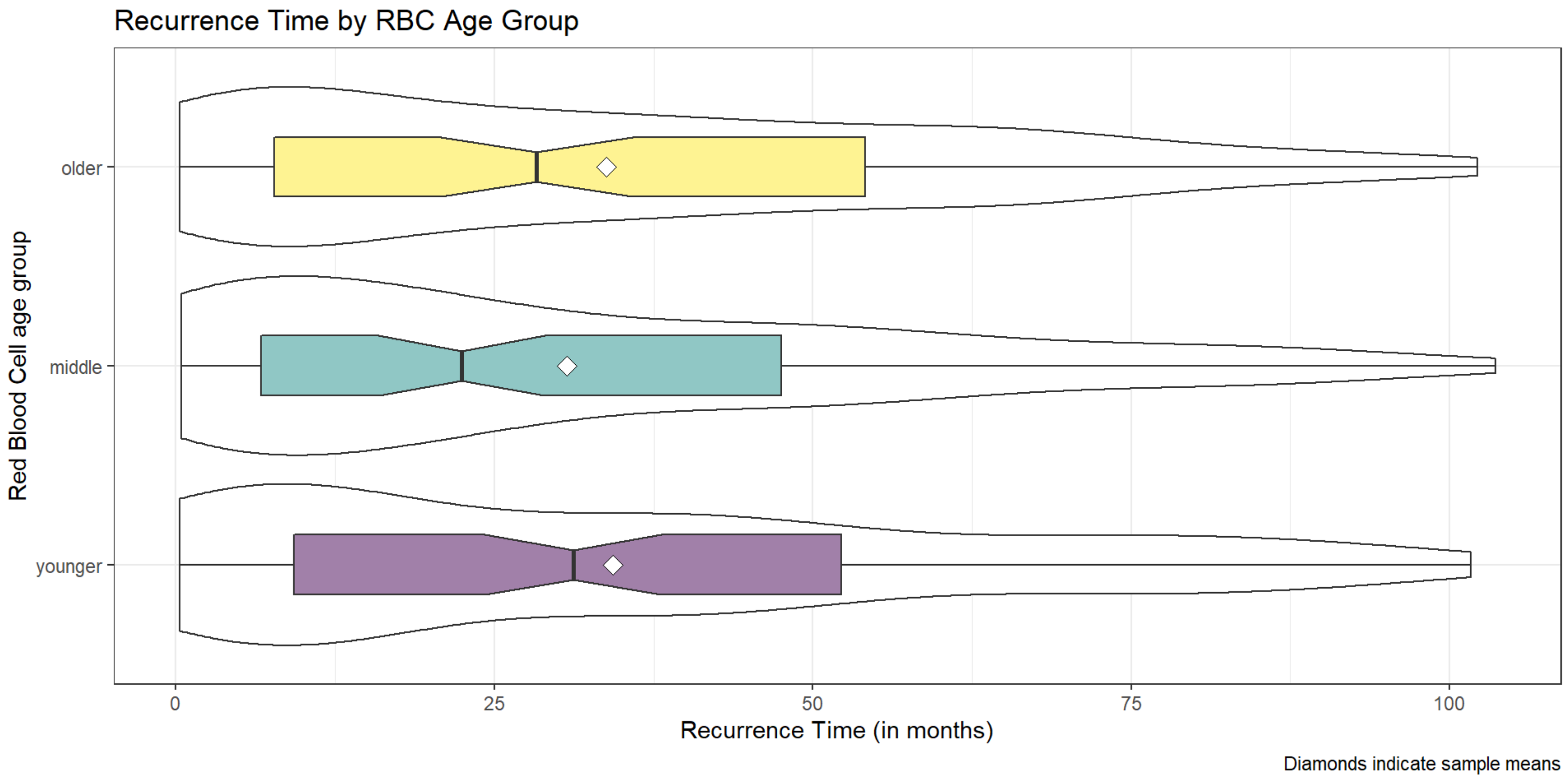
Normal Q-Q plot of m1 Residuals

```
ggplot(data = m1_aug, aes(sample = .resid)) +
  geom_qq() + geom_qq_line(col = "red")
```



Back to our Comparison Boxplot

- Does comparing means make sense here?
- Are these sample distributions "Normal-ish"?



Would a Transformation Help Us?

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

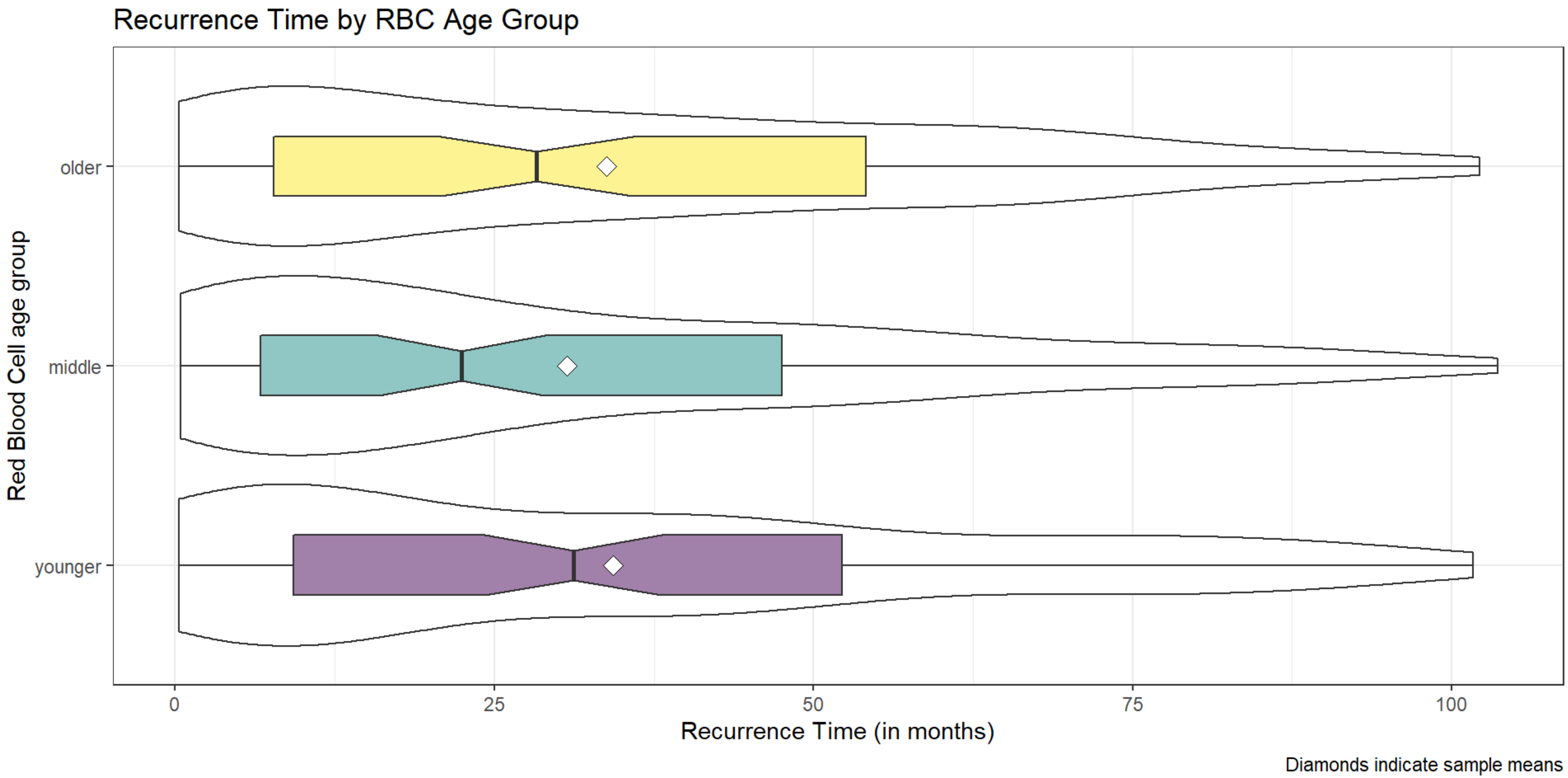
```
  min   Q1 median    Q3   max     mean       sd   n missing
 0.27 7.6    25.3 52.07 103.6 32.89225 28.47644 289       0
```

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

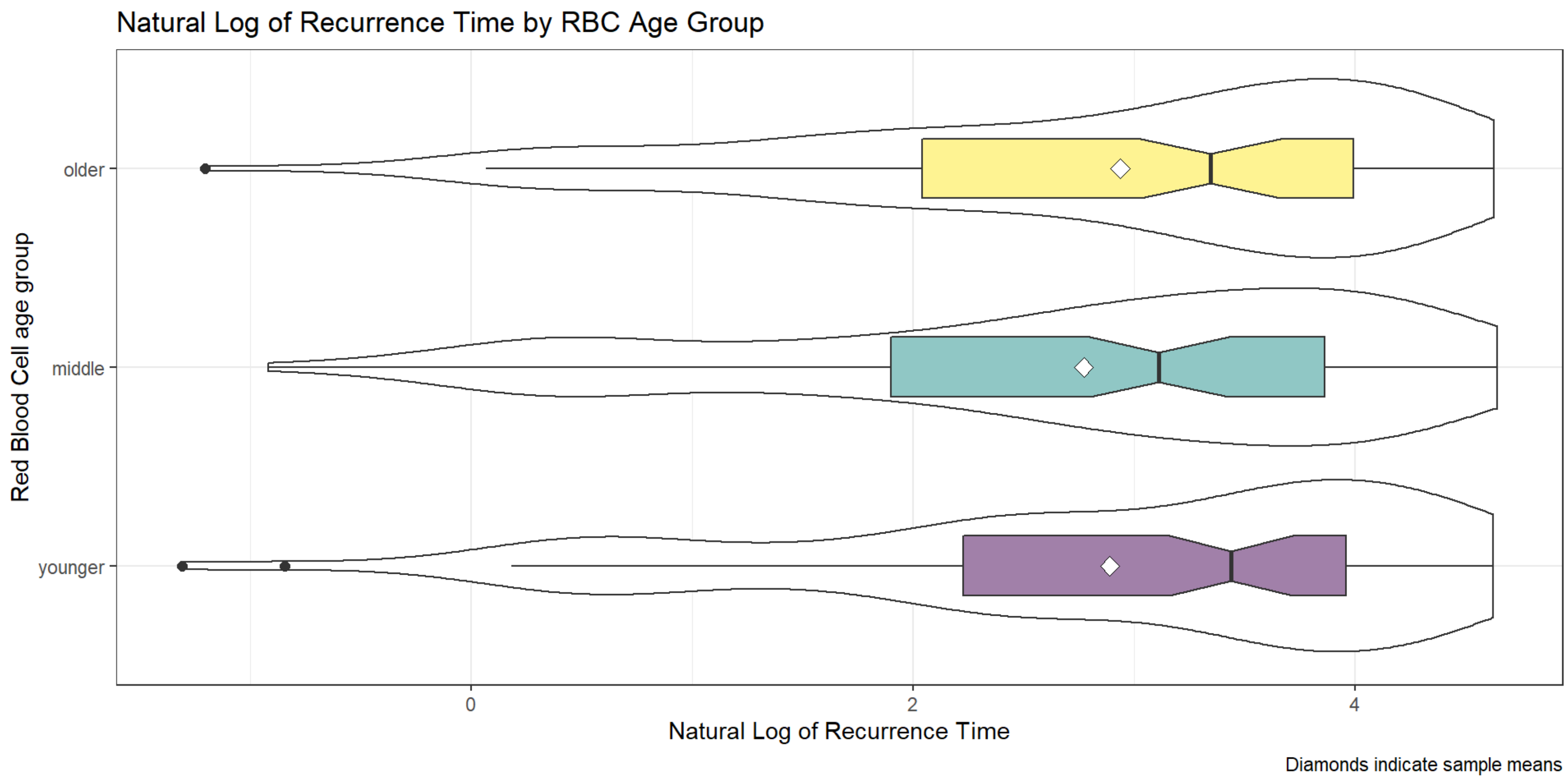
$log(time)$, or $1/time$, or $\sqrt{time}$, or $time^2$, for example…

What are we hoping these transformations will do?

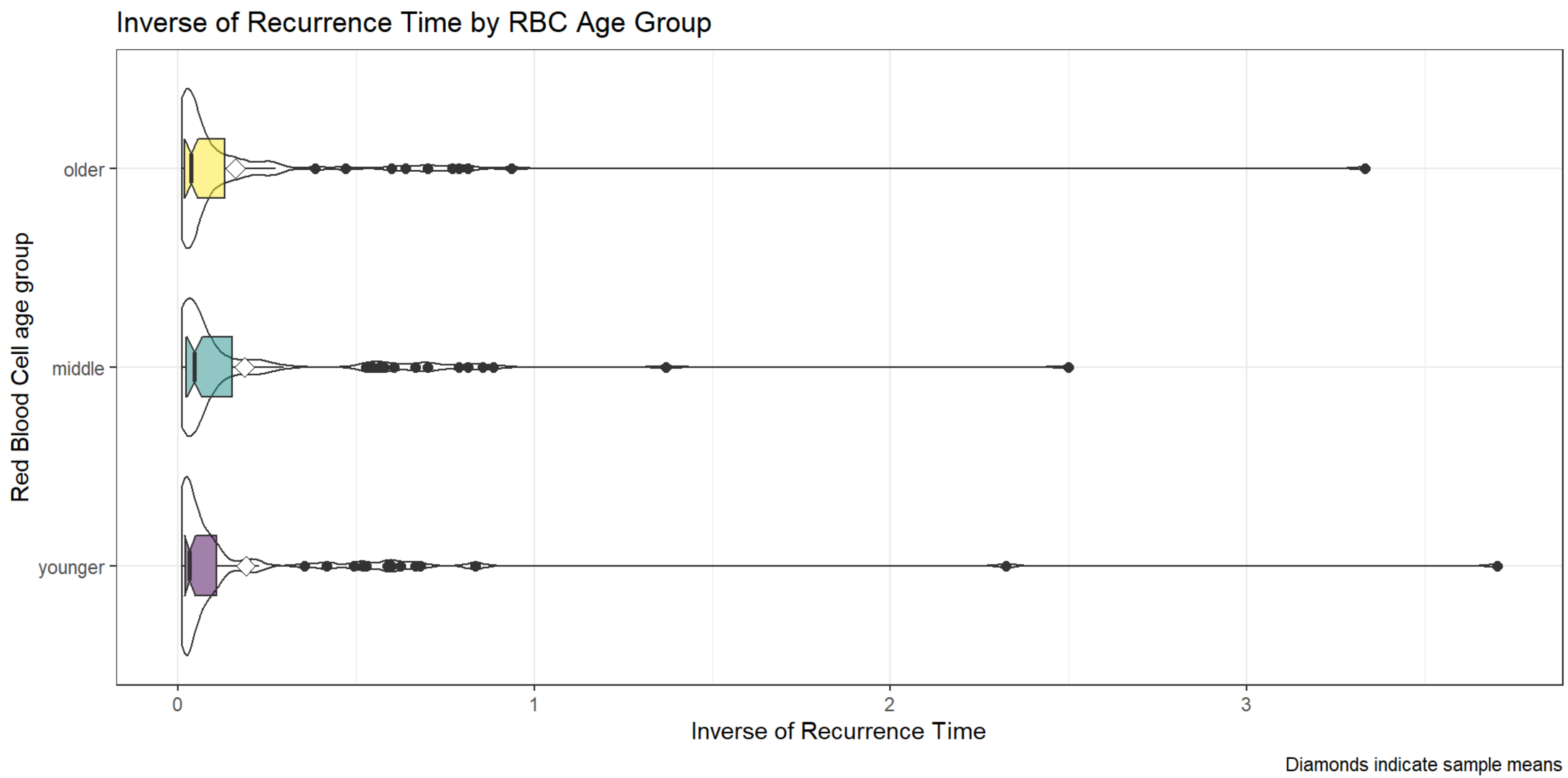
Boxplot 0: `recur_time` by `age_group`



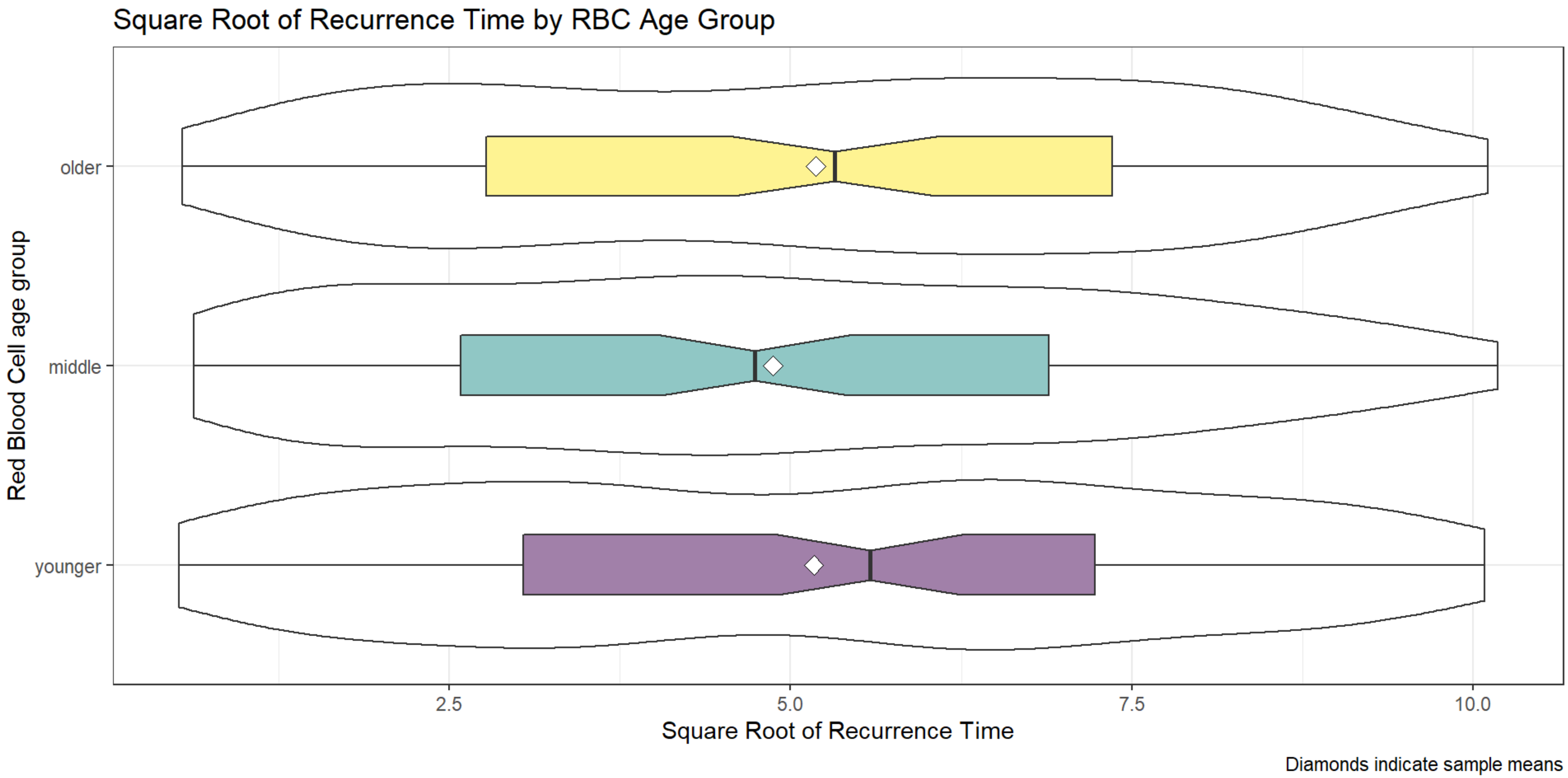
Boxplot 1: `log(recur_time)` by `age_group`



Boxplot 2: `1/(recur_time)` by `age_group`



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

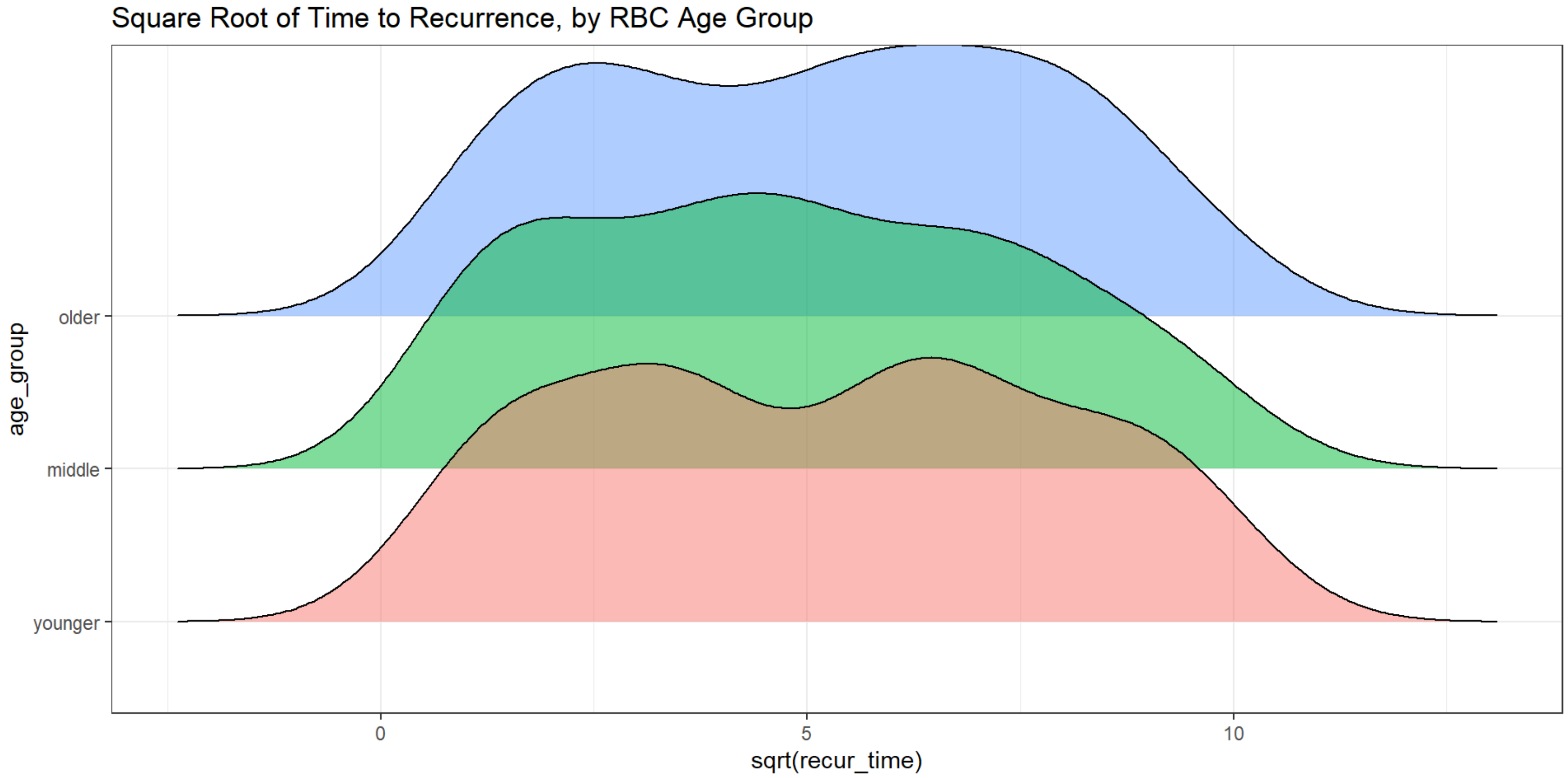


Code for Boxplot 3

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

Ridgeline Plot for $\sqrt{time}$?

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



Fit a Model to predict $\sqrt{time}$?

```r
m2 <- lm(sqrt(recur_time) ~ age_group, data = bs_cc)

extract_eq(m2, use_coefs = TRUE, wrap = TRUE, coef_digits = 3,
           terms_per_line = 1, operator_location = "start",
           font_size = "small")
```

$$
\begin{aligned}
\operatorname{\widehat{sqrt(recur\_time)}} &= 5.17 \\
&\quad - 0.299(\operatorname{age\_group}_{\operatorname{middle}}) \\
&\quad + 0.014(\operatorname{age\_group}_{\operatorname{older}})
\end{aligned}
$$

Predicted Values using m2

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

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

Predicted `recur_time` using `m2`

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

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

```
tidy(m2, conf.int = TRUE, conf.level = 0.90) |>
  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

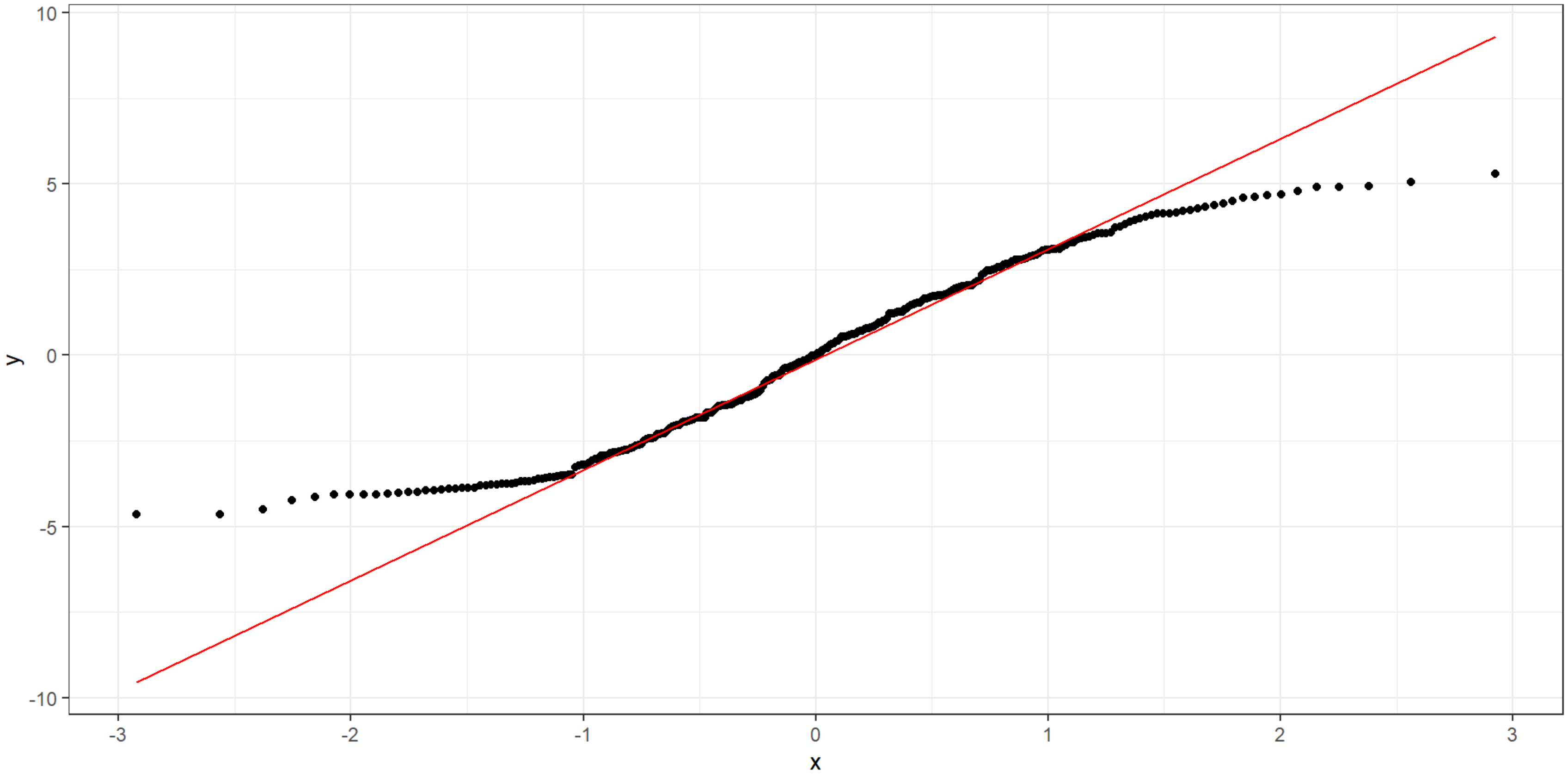
glance to summarize m2’s fit

```
glance(m2) |>
  select(r.squared, AIC, nobs, df, df.residual) |>
  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

Normal Q-Q plot of residuals for m2

```r
m2_aug <- augment(m2, data = bs_cc)

ggplot(data = m2_aug, aes(sample = .resid)) +
  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

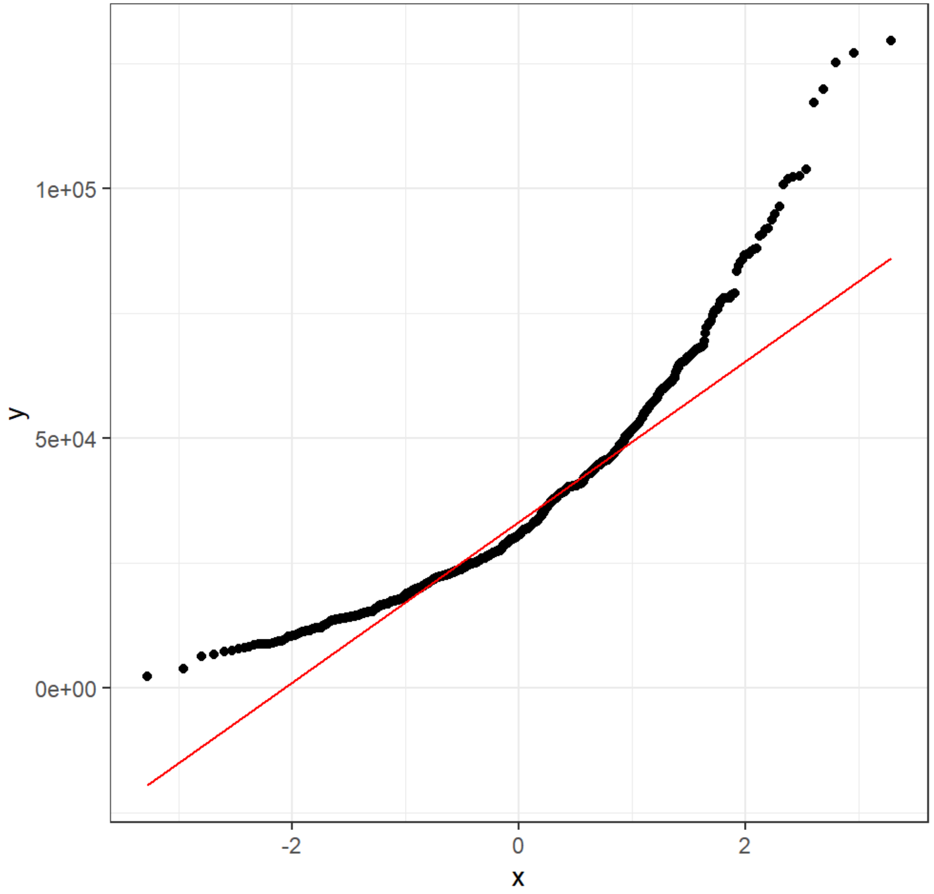
Consider the `n_income` data in `dm1000`

```r
dm1000 <- read_rds("c09/data/dm_1000.Rds")

mosaic::favstats(~ n_income, data = dm1000) |>
  select(n, missing, min, median, mean, max) |>
  kbl(digits = 2) |>
  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`

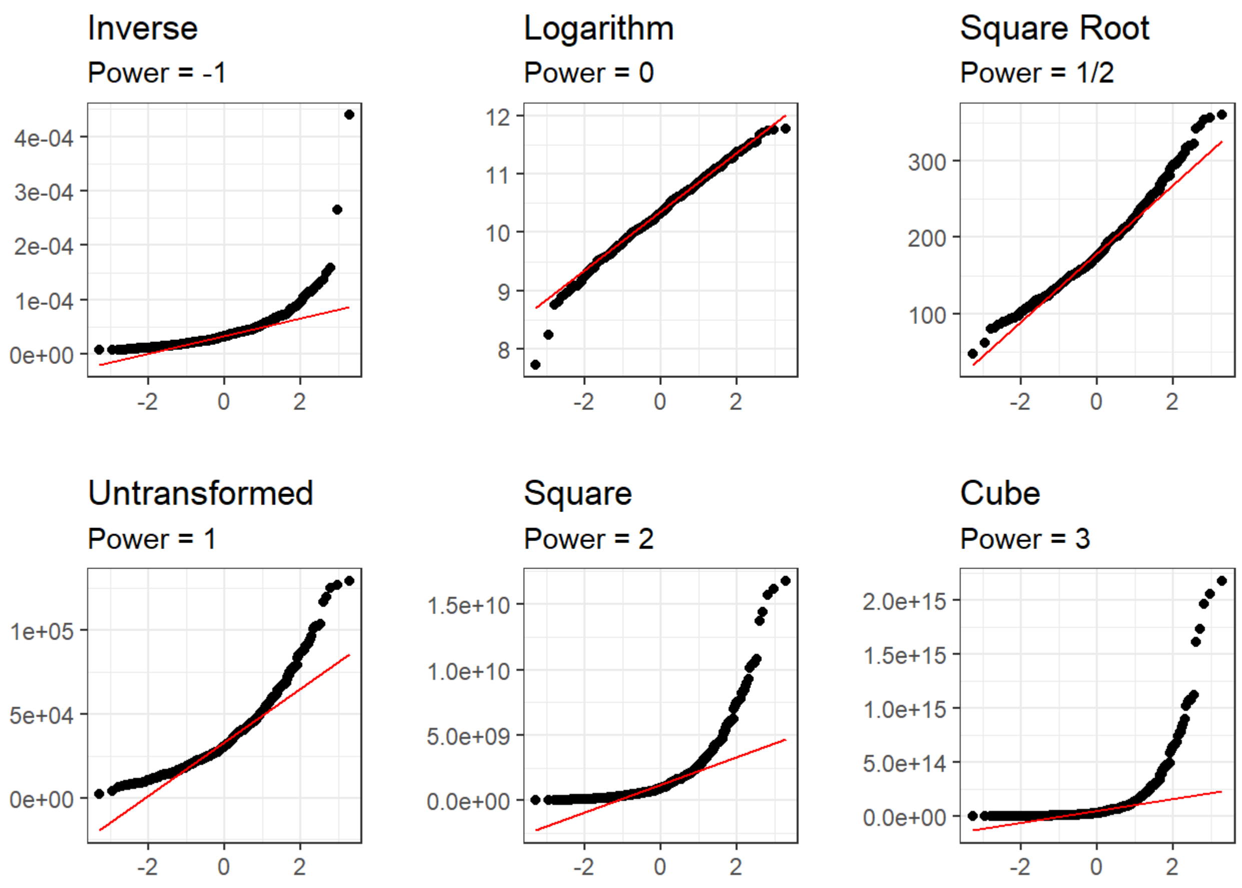
```r
dm972 <- dm1000 |> filter(complete.cases(n_income))
ggplot(data = dm972, aes(sample = n_income)) +
  geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1)
```



Ladder of `n_income` transformations

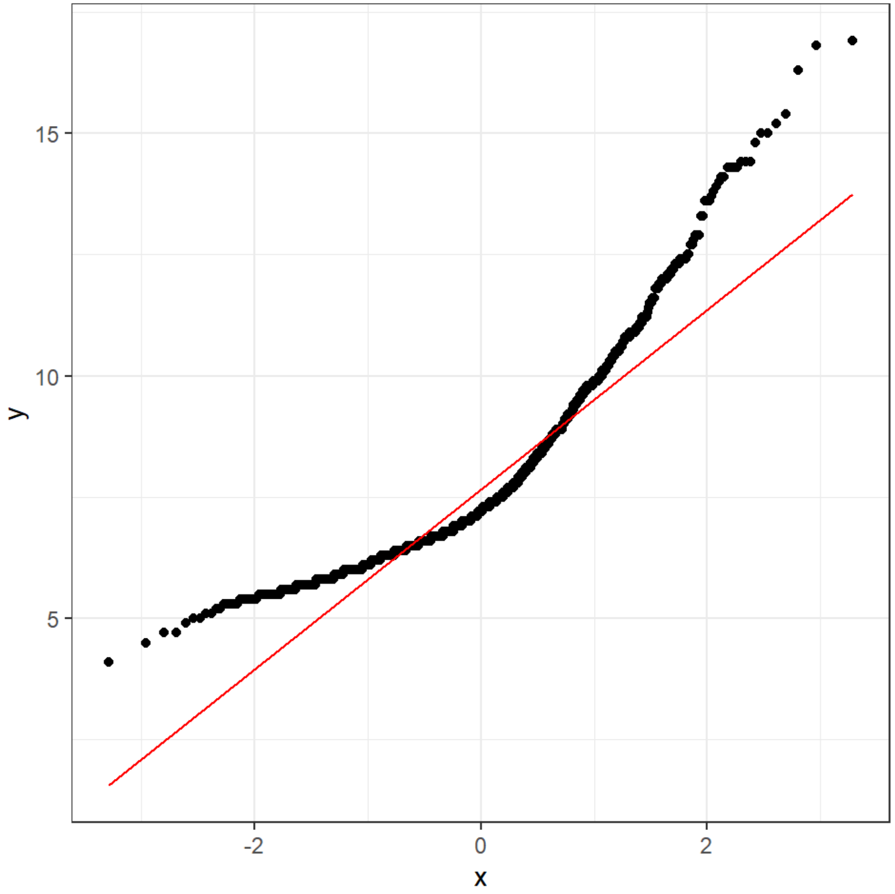
```r
p1 <- ggplot(data = dm972, aes(sample = n_income)) +
  geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1) +
  labs(title = "Untransformed", subtitle = "Power = 1", x = "", y = "")

p2 <- ggplot(data = dm972, aes(sample = n_income^2)) +
  geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1) +
  labs(title = "Square", subtitle = "Power = 2", x = "", y = "")

p3 <- ggplot(data = dm972, aes(sample = n_income^3)) +
  geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1) +
  labs(title = "Cube", subtitle = "Power = 3", x = "", y = "")

p4 <- ggplot(data = dm972, aes(sample = sqrt(n_income))) +
  geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1) +
  labs(title = "Square Root", subtitle = "Power = 1/2", x = "", y = "")

p5 <- ggplot(data = dm972, aes(sample = log(n_income))) +
  geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1) +
  labs(title = "Logarithm", subtitle = "Power = 0", x = "", y = "")
```

Ladder of `n_income` transformations



Hemoglobin A1c data in `dm1000`

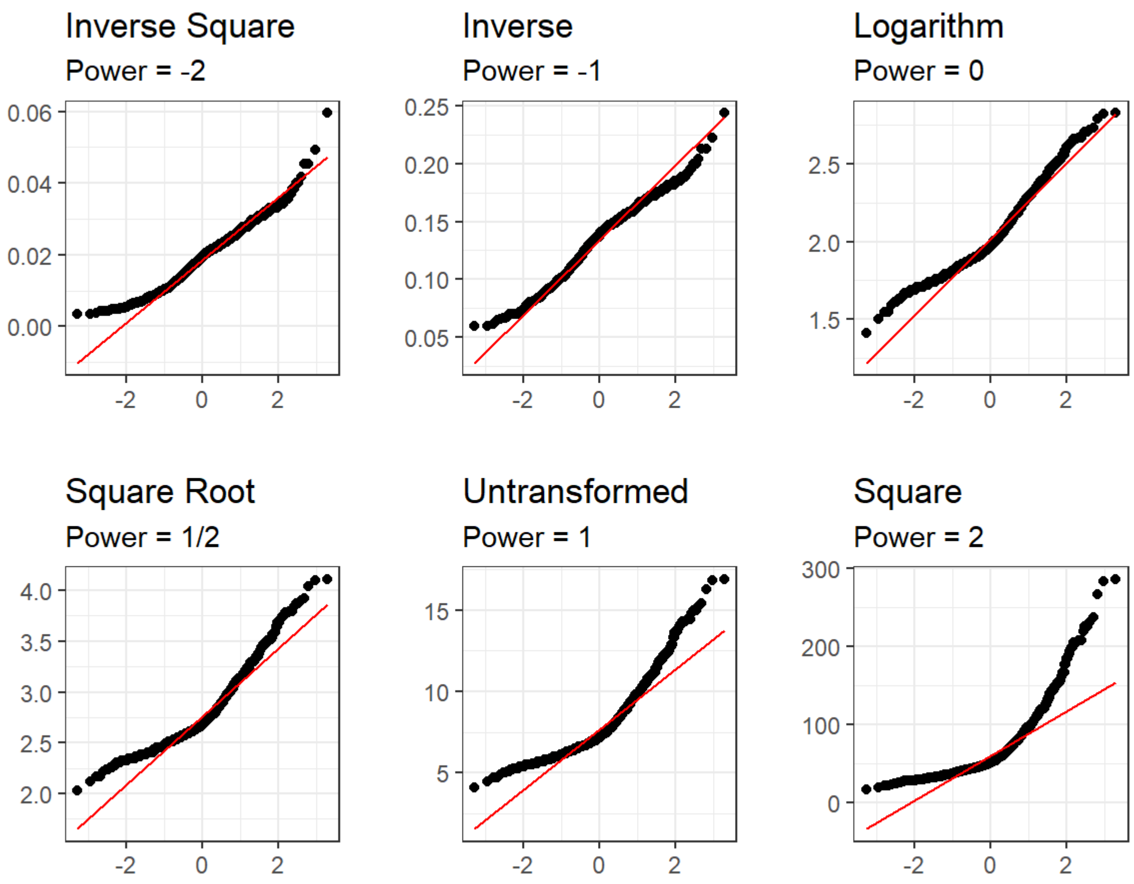
```r
dm985 <- dm1000 |> filter(complete.cases(a1c))
ggplot(data = dm985, aes(sample = a1c)) +
  geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1)
```



Ladder of A1c transformations

```r
p1 <- ggplot(data = dm985, aes(sample = a1c)) +
  geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1) +
  labs(title = "Untransformed", subtitle = "Power = 1", x = "", y = "")

p2 <- ggplot(data = dm985, aes(sample = a1c^2)) +
  geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1) +
  labs(title = "Square", subtitle = "Power = 2", x = "", y = "")

p3 <- ggplot(data = dm985, aes(sample = a1c^3)) +
  geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1) +
  labs(title = "Cube", subtitle = "Power = 3", x = "", y = "")

p4 <- ggplot(data = dm985, aes(sample = sqrt(a1c))) +
  geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1) +
  labs(title = "Square Root", subtitle = "Power = 1/2", x = "", y = "")

p5 <- ggplot(data = dm985, aes(sample = log(a1c))) +
  geom_qq() + geom_qq_line(col = "red") + theme(aspect.ratio = 1) +
  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.

```r
bs_dat2 <- bs_dat |>
  filter(complete.cases(recur_time, units))

bs_dat2 |> tabyl(units)
```

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

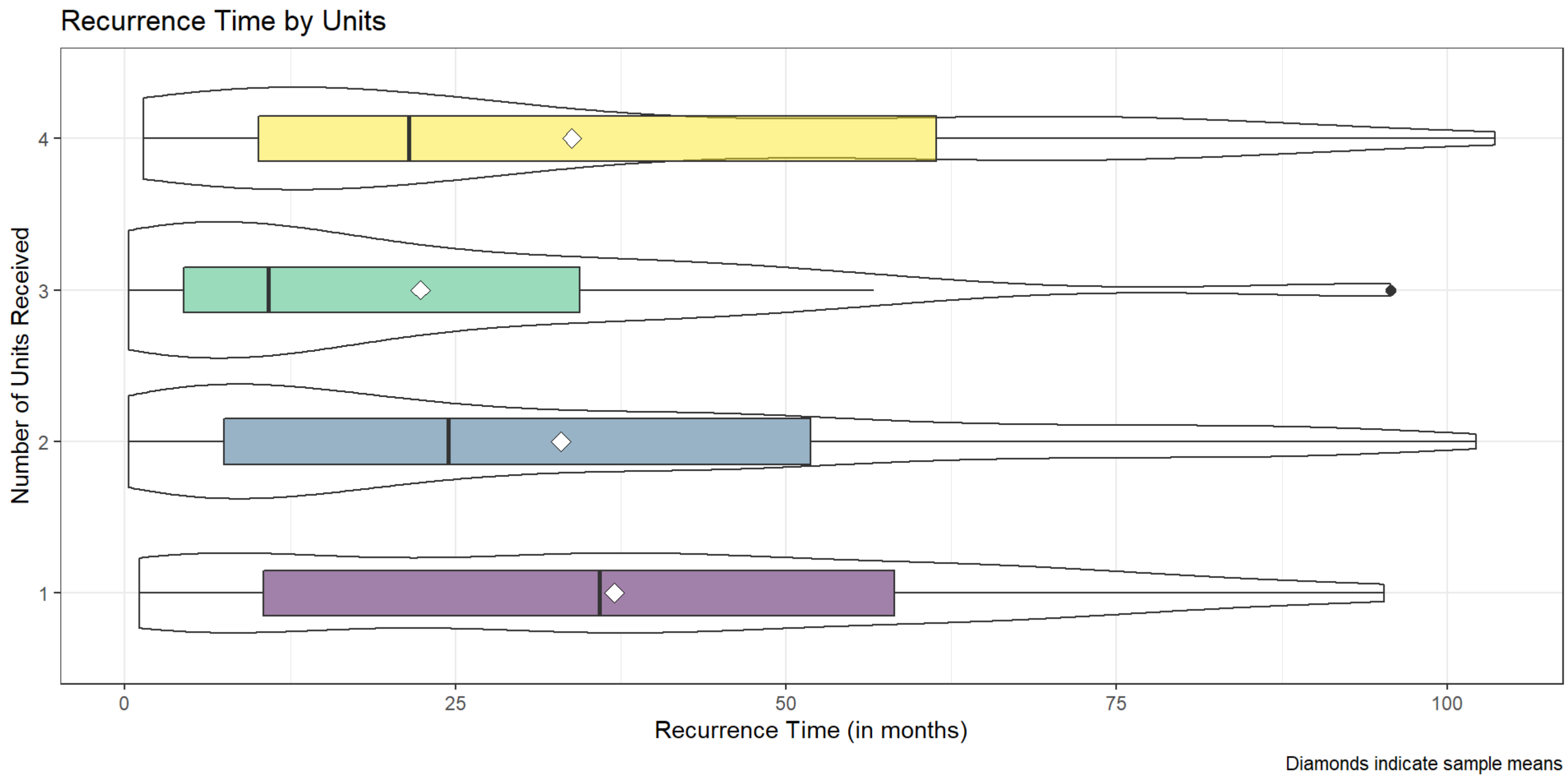
Scatterplot of recur_time vs. age_group

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

Comparison Boxplot

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

Comparison Boxplot



Model Time using Units

```
m3 <- lm(recur_time ~ units, data = bs_dat2)

extract_eq(m3, use_coefs = TRUE, coef_digits = 2)
```

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

```
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
```

Model Square Root of Time using Units

```r
m4 <- lm(sqrt(recur_time) ~ units, data = bs_dat2)

extract_eq(m4, use_coefs = TRUE, coef_digits = 2)
```

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

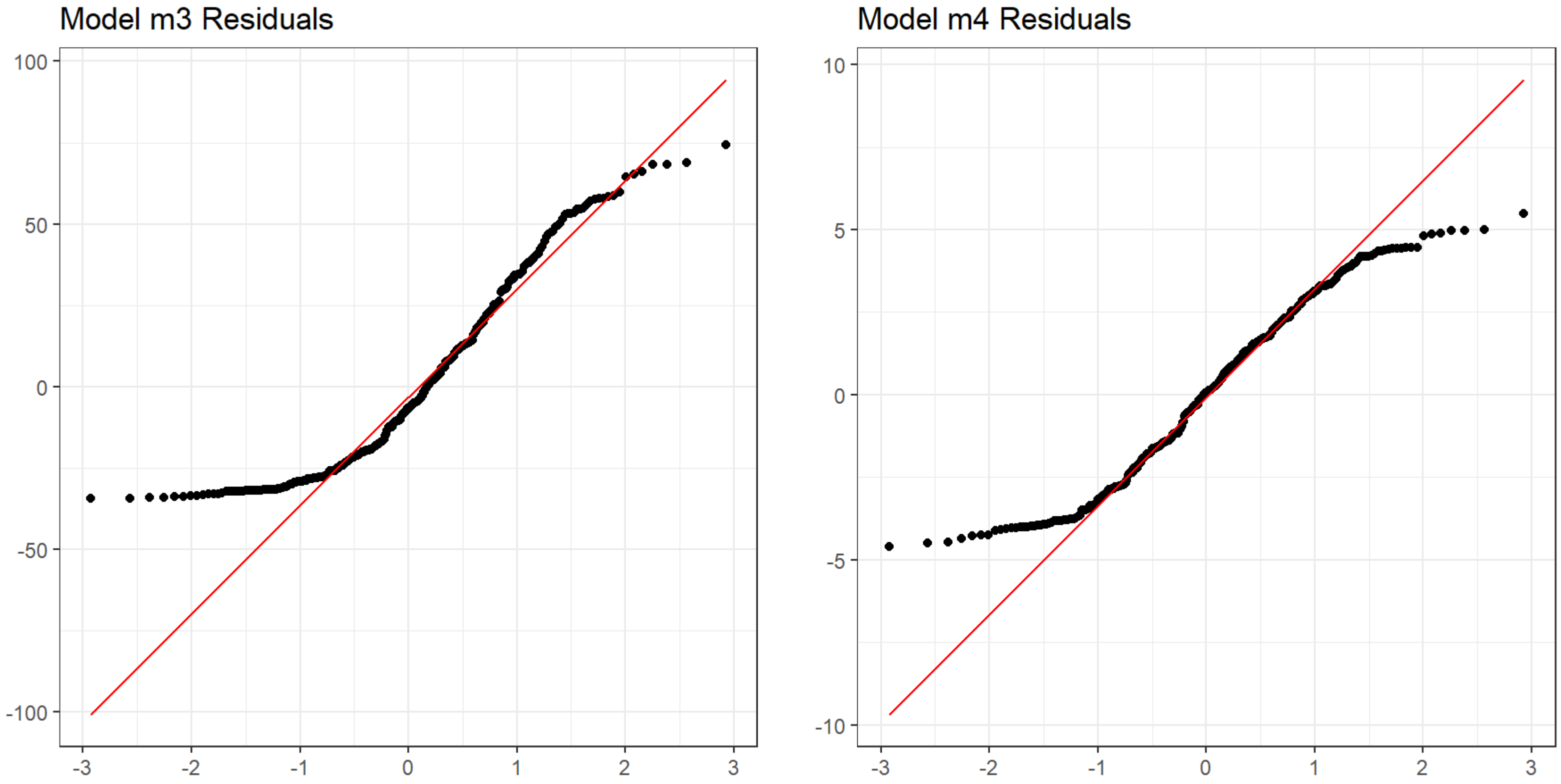
```r
tidy(m4, conf.int = TRUE, conf.level = 0.90)
```

```
# 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
```

Normal Q-Q plots of Residuals

```r
m3_aug <- augment(m3, data = bs_dat2)
m4_aug <- augment(m4, data = bs_dat2)

p1 <- ggplot(m3_aug, aes(sample = .resid)) +
  geom_qq() + geom_qq_line(col = "red") +
  theme(aspect.ratio = 1) +
  labs(title = "Model m3 Residuals", x = "", y = "")

p2 <- ggplot(m4_aug, aes(sample = .resid)) +
  geom_qq() + geom_qq_line(col = "red") +
  theme(aspect.ratio = 1) +
  labs(title = "Model m4 Residuals", x = "", y = "")

p1 + p2
```

Normal Q-Q plots of Residuals



Compare fits of `m1` and `m3`?

```
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
```

```
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

```r
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
```