431 Class 08

Thomas E. Love, Ph.D.

2022-09-22

Today's Agenda

- Working again with the `dm1000` data from Class 07
- Confidence Intervals around a Mean
- Building Visualizations to Compare Distributions
- Confidence Intervals for a Difference between Means

Today's Packages

```
 1 library(broom)      ## tidying model output
 2 library(ggridges)   ## help building ridgeline plots
 3 library(Hmisc)      ## smean.cl.boot() and smean.cl.norm()
 4 library(janitor)
 5 library(kableExtra) ## for table neatening
 6 library(naniar)
 7 library(patchwork)
 8 library(readxl)     ## new today, read in .xls or .xlsx files
 9 library(tidyverse)
10
11 theme_set(theme_bw())
```

Data Ingest

Today, we'll use an Excel file (.xls, rather than .csv) to import the `dm1000` data.

```
dm1000 <- read_excel("c08/data/dm_1000.xls") |>
  clean_names() |>
  mutate(across(where(is.character), as_factor)) |>
  mutate(subject = as.character(subject))
```

- The `readxl` package is a non-core part of the tidyverse, and also includes functions called `read_xls()` and `read_xlsx()`.
- Visit https://readxl.tidyverse.org/.

The dm1000 tibble

```
1 dm1000
```

```
# A tibble: 1,000 × 17
   subject   age insurance  n_income    ht    wt   sbp   dbp   a1c   ldl
tobacco
   <chr>   <dbl> <fct>         <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <fct>
 1 M-0001     55 Medicaid      29853  1.63 103.    145    70   6.4   221
Current
 2 M-0002     52 Commercial    31248  1.75 112.    151    77   8.5   116 Never
 3 M-0003     69 Medicare      23362  1.65  74.9   127    73   8.9    52
Former
 4 M-0004     57 Medicaid      26033  1.63  81.4   125    74   6.8   122 Never
 5 M-0005     68 Medicare      85374  1.69  92.6   120    73  10.3    94 Never
 6 M-0006     56 Medicaid      31273  1.71  54.6   127    75  12.3    NA
Current
 7 M-0007     54 Commercial    25445  1.68  81.6   114    81   6.5   100
Current
```

Estimating a Population Mean from a Sample

Suppose our sample in `dm1000` is a random sample from the population of all Cuyahoga County residents between the ages of 31-75 receiving care for diabetes.

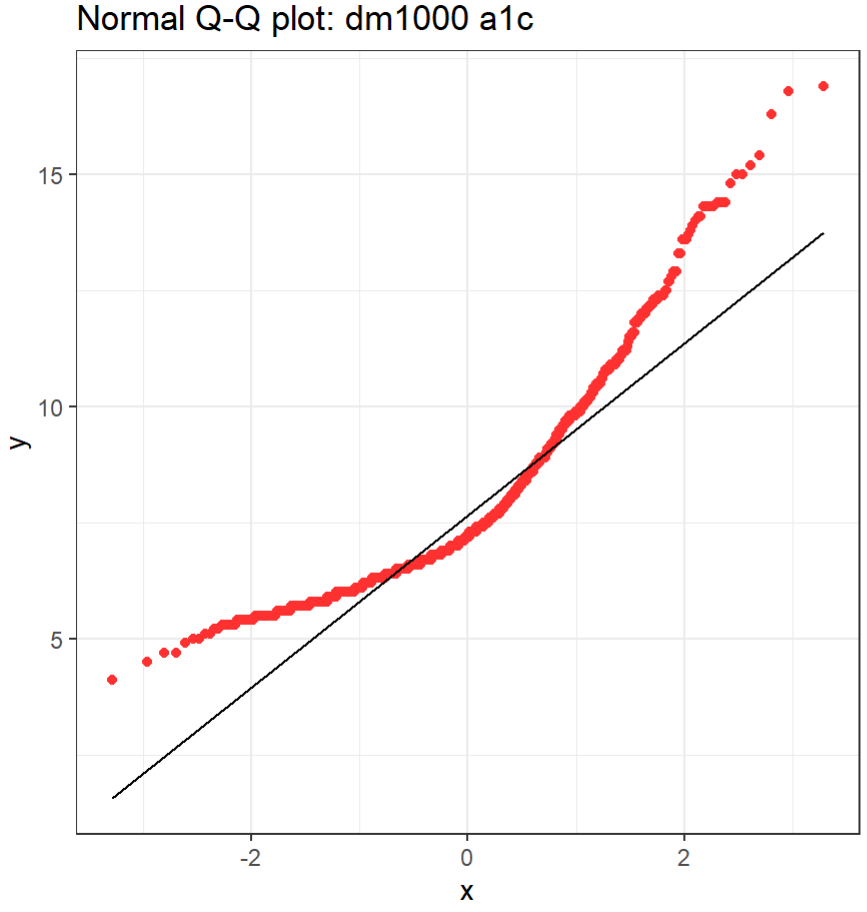
What's a good estimate for the mean Hemoglobin A1c of the people in that population?

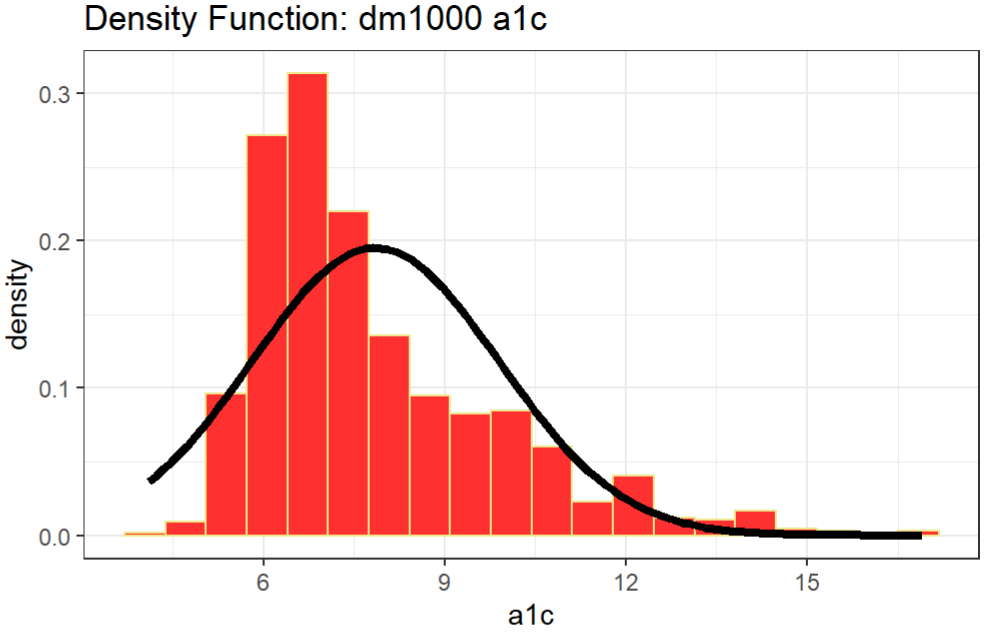
- How would we make this estimate using our data?
- What would we want to know about the data?
- DTDP

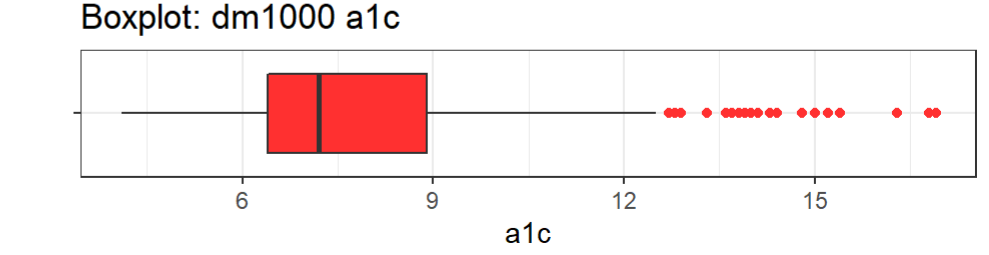
Hemoglobin A1c in the dm1000 sample

```
p1 <- ggplot(dm1000, aes(sample = a1c)) +
  geom_qq(col = "firebrick1") +
  geom_qq_line(col = "black") +
  theme(aspect.ratio = 1) +
  labs(title = "Normal Q-Q plot: dm1000 a1c")

p2 <- ggplot(dm1000, aes(x = a1c)) +
  geom_histogram(aes(y = stat(density)),
                 bins = 20, fill = "firebrick1", col = "khaki") +
  stat_function(fun = dnorm,
                args = list(mean = mean(dm1000$a1c, na.rm = TRUE),
                            sd = sd(dm1000$a1c, na.rm = TRUE)),
                col = "black", lwd = 1.5) +
  labs(title = "Density Function: dm1000 a1c")

p3 <- ggplot(dm1000, aes(x = a1c, y = "")) +
  geom_boxplot(fill = "firebrick1", outlier.color = "firebrick1") +
  labs(title = "Boxplot: dm1000 a1c", y = "")
```

Hemoglobin A1c in the `dm1000` sample







Numerical Summaries of A1c in dm1000

- Should we assume the A1c values are drawn from a Normal distribution?

```
1 mosaic::favstats(~ a1c, data = dm1000) |>
2   kbl(digits = 2) |>
3   kable_styling(font_size = 32, full_width = FALSE)
```

min	Q1	median	Q3	max	mean	sd	n	missing
4.1	6.4	7.2	8.9	16.9	7.85	2.04	985	15

- The `kbl()` and `kable_styling()` functions are from the `kableExtra` package.
  - Read more about kableExtra in its vignette.

What if we assume the A1cs are Normally distributed?

Then we could use a linear regression model to obtain a 95% confidence interval for the population mean.

```
m1 <- lm(a1c ~ 1, data = dm1000)
tidy(m1, conf.int = TRUE, conf.level = 0.95) |>
  select(estimate, conf.low, conf.high)
```

```
# A tibble: 1 × 3
  estimate conf.low conf.high
     <dbl>    <dbl>     <dbl>
1     7.85     7.73      7.98
```

What is the model we've fit here?

```
1 m1
```

```
Call:
lm(formula = a1c ~ 1, data = dm1000)

Coefficients:
(Intercept)
      7.853
```

- This "intercept only" model simply predicts the mean value of our outcome, `a1c`.

Interpreting this interval? (1/3)

- Our 95% confidence interval for the population mean A1c is (7.73, 7.98).

We are estimating the mean of the **population** based on a mean from our *sample* of 1000 observations taken (at random, we assume) from that population.

- Sadly, this **doesn't** mean we're 95% confident that the actual population mean is in that range, even though lots and lots of people (incorrectly) assume it does.

Interpreting this interval? (2/3)

- Our 95% confidence interval for the population mean A1c is (7.73, 7.98).

Essentially, we have 95% confidence in the **process** of fitting confidence intervals this way. If we fit 100 such confidence intervals to a variety of data sets, we have some reason to anticipate that 95 of them will contain the actual unknown value of the population mean.

Interpreting this interval? (3/3)

- Our 95% confidence interval for the population mean A1c is (7.73, 7.98).

That's oversimplifying a little, and a particular concern in this case is that the data are somewhat skewed (at any rate, not very close to Normally distributed) and this may impact our ability to generate accurate and efficient confidence intervals via a linear model like this.

An Equivalent Approach

We could use a t test to obtain a 95% confidence interval for the population mean.

```
1 tt <- t.test(dm1000$a1c, conf.level = 0.95)
2 tidy(tt) |> select(estimate, conf.low, conf.high)
```

```
# A tibble: 1 × 3
  estimate conf.low conf.high
     <dbl>    <dbl>     <dbl>
1     7.85     7.73      7.98
```

- This is exactly the same result as we obtain from the linear model `m1`.

Another Equivalent Approach

We could use a function called `smean.cl.normal()` from the `Hmisc` package to obtain the same 95% confidence interval for the population mean.

```
1 smean.cl.normal(dm1000$a1c, conf.int = 0.95)
```

```
    Mean     Lower     Upper 
7.852893 7.725167 7.980619 
```

- Again, this is the same result as we have seen previously.

What if we weren't willing to assume that the A1c values came from a Normal distribution?

- Use the bootstrap to estimate the population mean with 95% confidence.

```
set.seed(2022431)  # why do we set a seed here?
smean.cl.boot(dm1000$a1c, conf.int = 0.95)
```

```
    Mean     Lower     Upper 
7.852893 7.726066 7.979513 
```

Bootstrap CI with a different seed

```
1 set.seed(1234567)  # what happens if we change the seed
2 smean.cl.boot(dm1000$a1c, conf.int = 0.95)
```

```
    Mean    Lower    Upper
7.852893 7.725378 7.978716
```

- The `smean.cl.boot()` function comes from the `Hmisc` package.

> Bootstrap is a resampling method where large numbers of samples of the same size are repeatedly drawn, with replacement, from a single original sample.

95% CIs for Population Mean A1c

Approach	Estimate	95% CI	Assume Normality?
Linear Model	7.853	(7.725, 7.981)	Yes
Bootstrap	7.853	(7.726, 7.978)	No

- How does this match up with our understanding of the distribution of the A1c data?
- What's an appropriate number of decimal places to use here?

```
1  dm1000 |> select(a1c) |> head(15) |> as.vector()
```

```
$a1c
 [1]  6.4  8.5  8.9  6.8 10.3 12.3  6.5  5.4  7.1  8.9  6.6  6.4 11.1 11.8  6.5
```

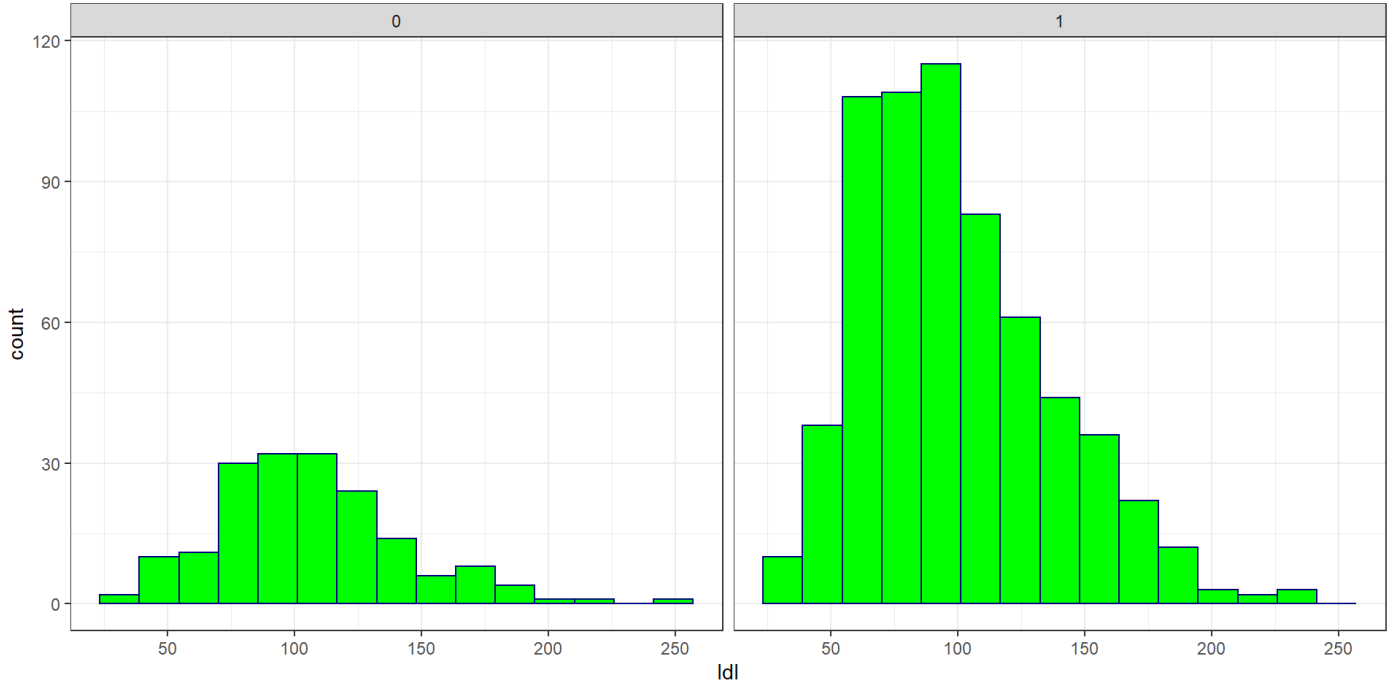
- Might the fairly large sample size (n = 985 non-missing) have something to do with these results?

Comparing Two Distributions

Is LDL higher or lower among adults with diabetes who have a statin prescription?

```
1 ggplot(data = dm1000, aes(x = ldl)) +
2   geom_histogram(bins = 15, fill = "green", col = "navy") +
3   facet_wrap(~ statin)
```

```
Warning: Removed 178 rows containing non-finite values (stat_bin).
```

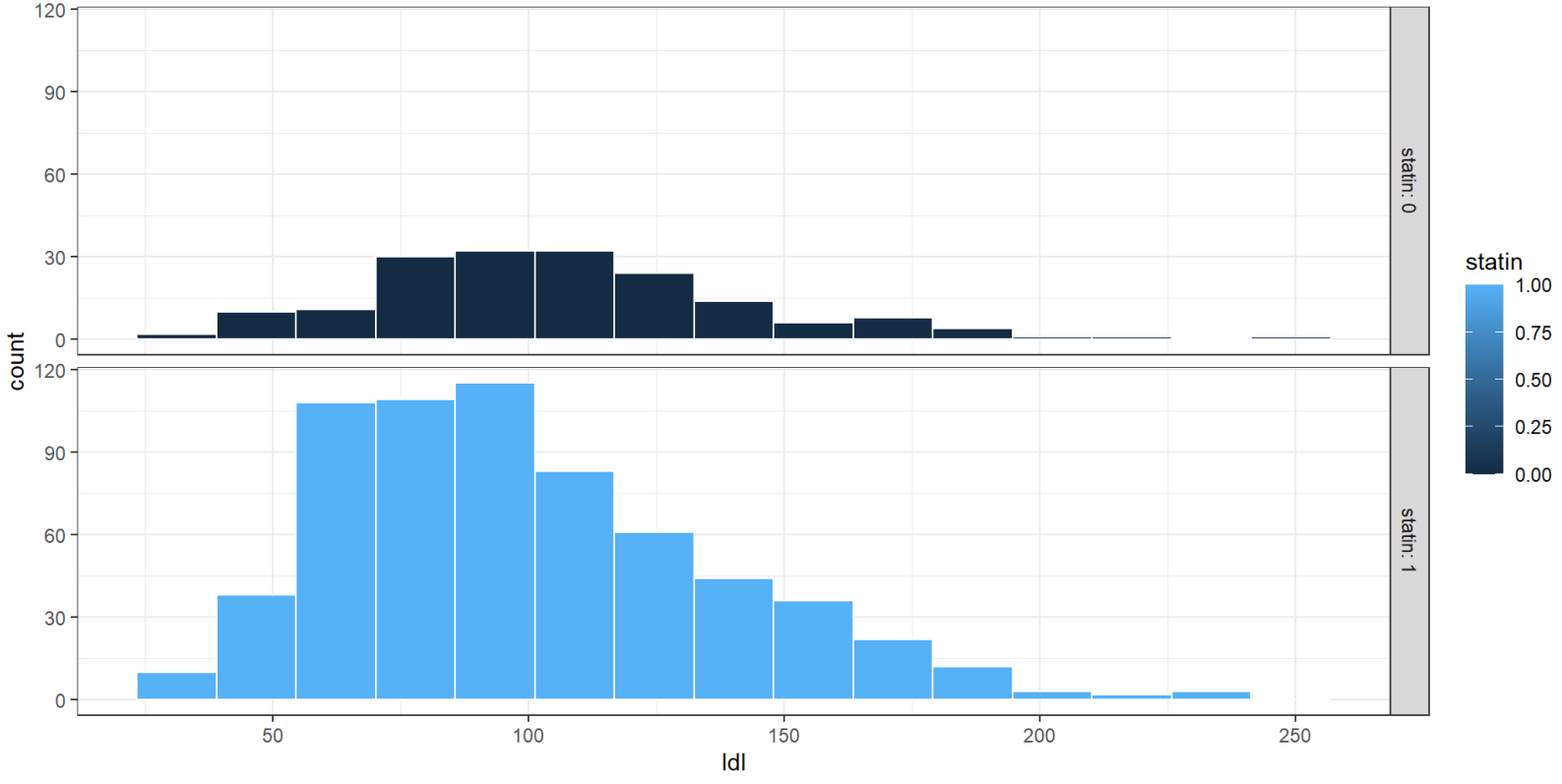


How might we improve this plot?

1. Remove the warning about non-finite (missing) values.
2. Place the histograms vertically to ease comparisons.
3. Fill the histograms differently for statin and no statin.
4. Augment the labels (0 and 1) to show they identify statin use.

LDL stratified by `statin` (Plot 2)

```
1  tempdat <- dm1000 |> filter(complete.cases(ldl, statin))
2
3  ggplot(data = tempdat, aes(x = ldl, fill = statin)) +
4    geom_histogram(bins = 15, col = "white") +
5    facet_grid(statin ~ ., labeller = "label_both")
```

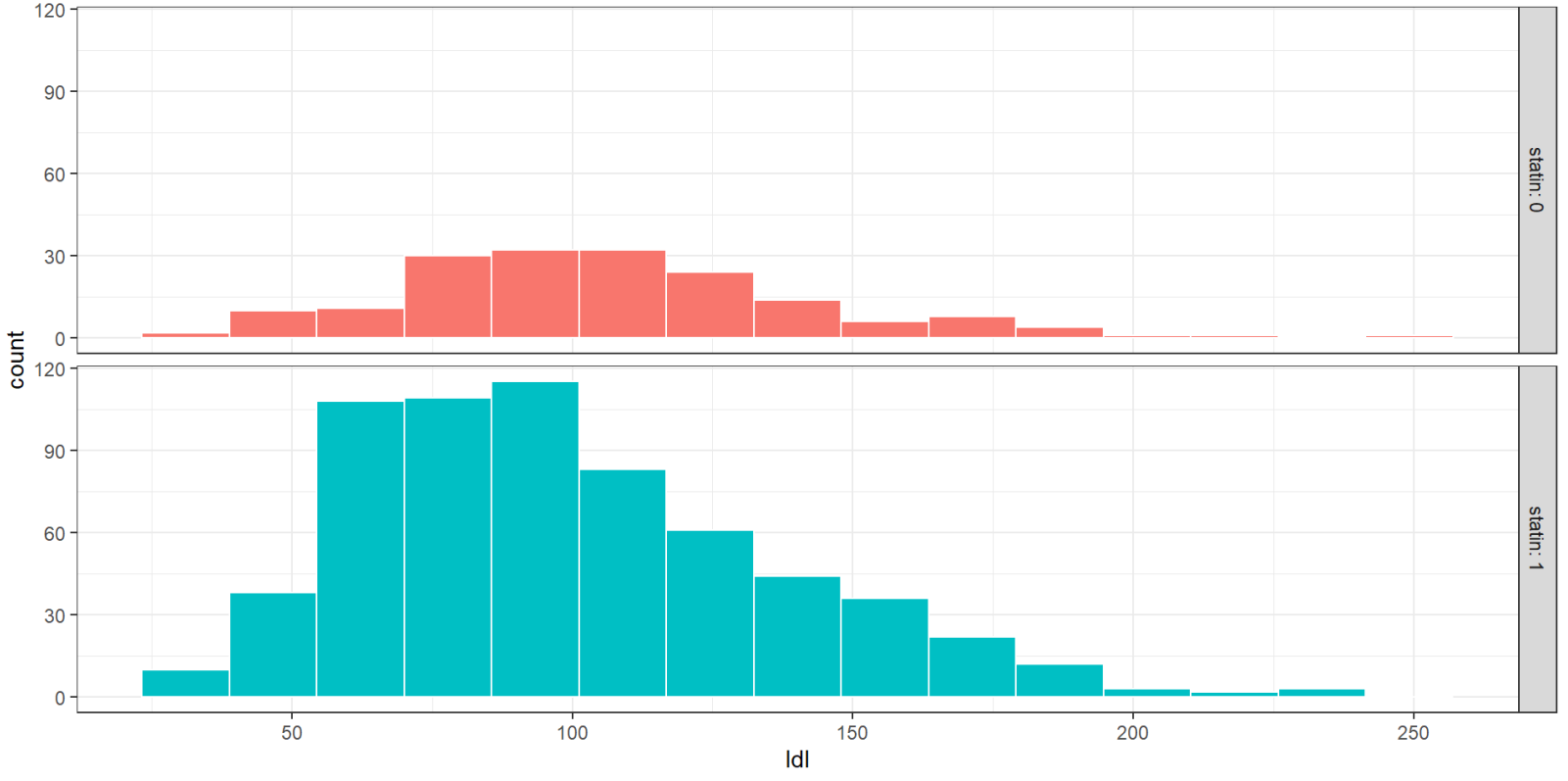


Problems with the previous plot

1. Statin is actually a two-category variable (1 and 0 are just codes) but the legend is treating it as if it was a numeric variable.
2. Do we actually need the legend (called a guide in R) or can we remove it?

But `statin` is categorical? (Plot 3)

```
1  tempdat <- dm1000 |> filter(complete.cases(ldl, statin))
2  ggplot(data = tempdat, aes(x = ldl, fill = factor(statin))) +
3    geom_histogram(bins = 15, col = "white") +
4    facet_grid(statin ~ ., labeller = "label_both") +
5    guides(fill = "none")
```



Faceting

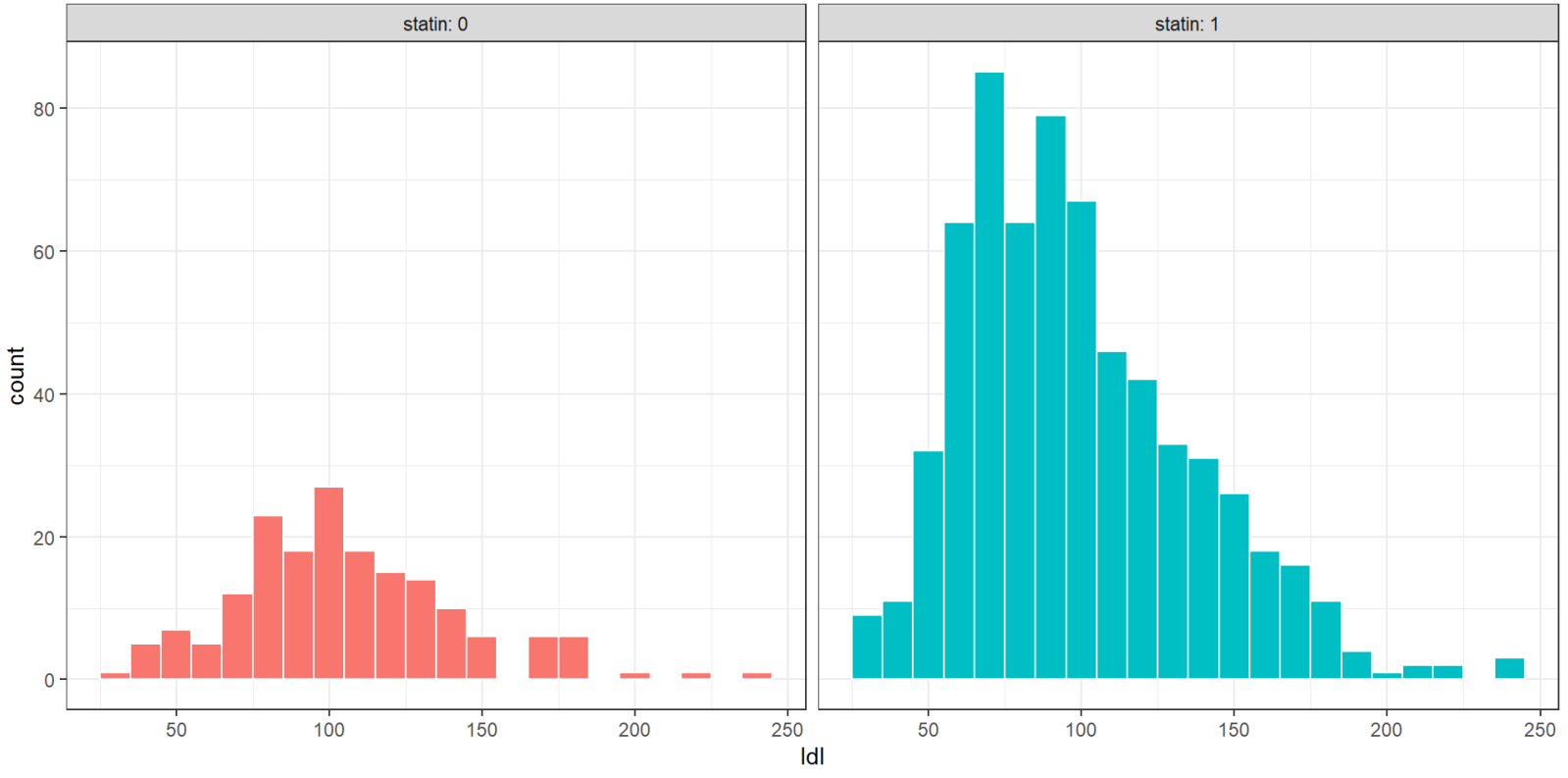
It's very useful to split data into groups and plot each group separately to make comparisons across the groups. We can then draw those subplots side by side.

We have two main tools: `facet_wrap()` and `facet_grid()`

- `facet_wrap(~ grp1)` to obtain plots within each `grp1` arranged into horizontal subpanels and wrapping around, like words on a page.
- `facet_grid(grp1 ~ .)` to obtain plots within each `grp1` arranged vertically (vertical subpanels)
- `facet_grid(grp1 ~ grp2)` to obtain plots within each combination of `grp1` and `grp2` with vertical and horizontal subpanels.

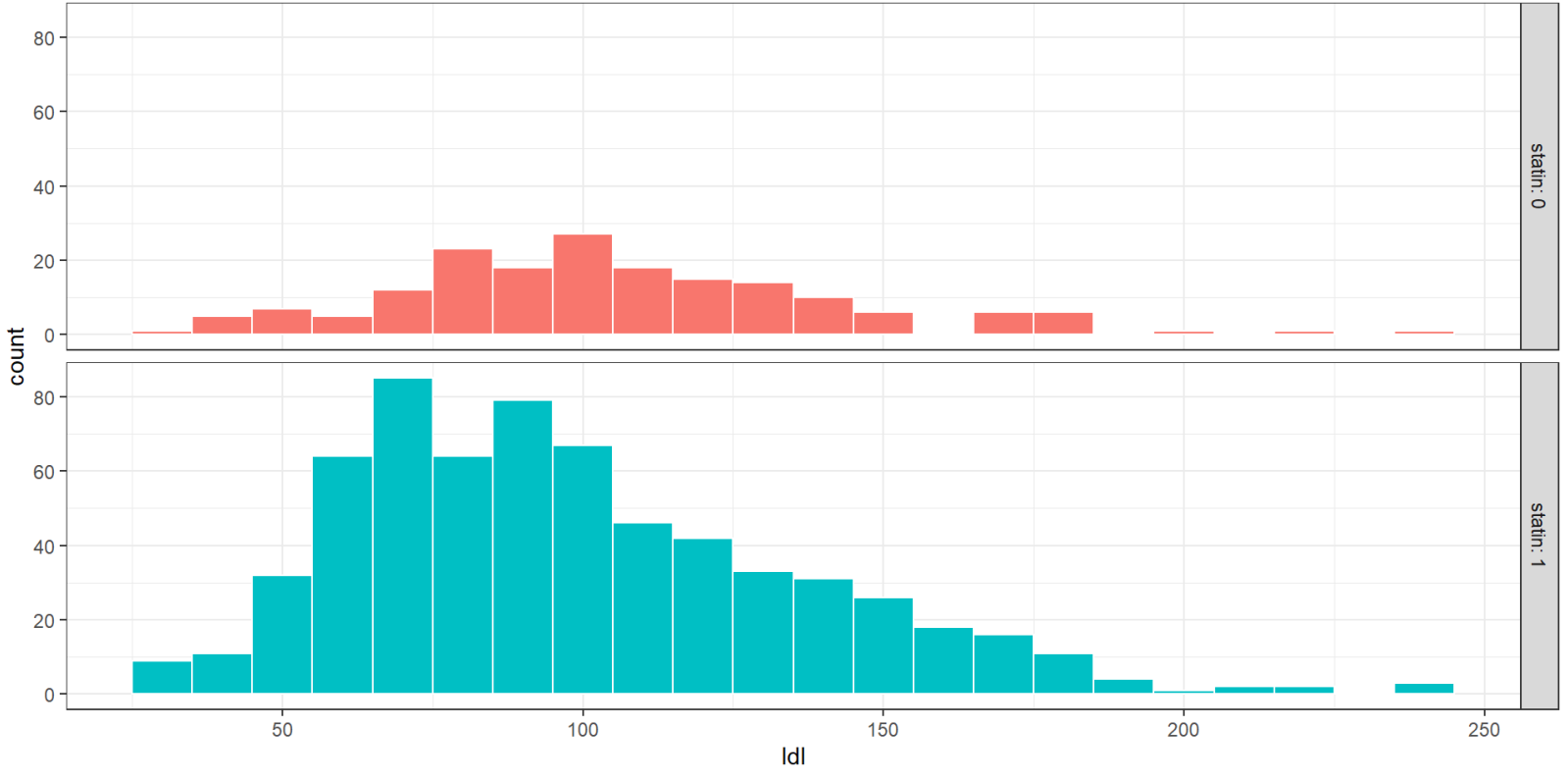
Using `facet_wrap()`

```
1  tempdat <- dm1000 |> filter(complete.cases(ldl, statin))
2  ggplot(data = tempdat, aes(x = ldl, fill = factor(statin))) +
3    geom_histogram(binwidth = 10, col = "white") +
4    facet_wrap(~ statin, labeller = "label_both") +
5    guides(fill = "none")
```



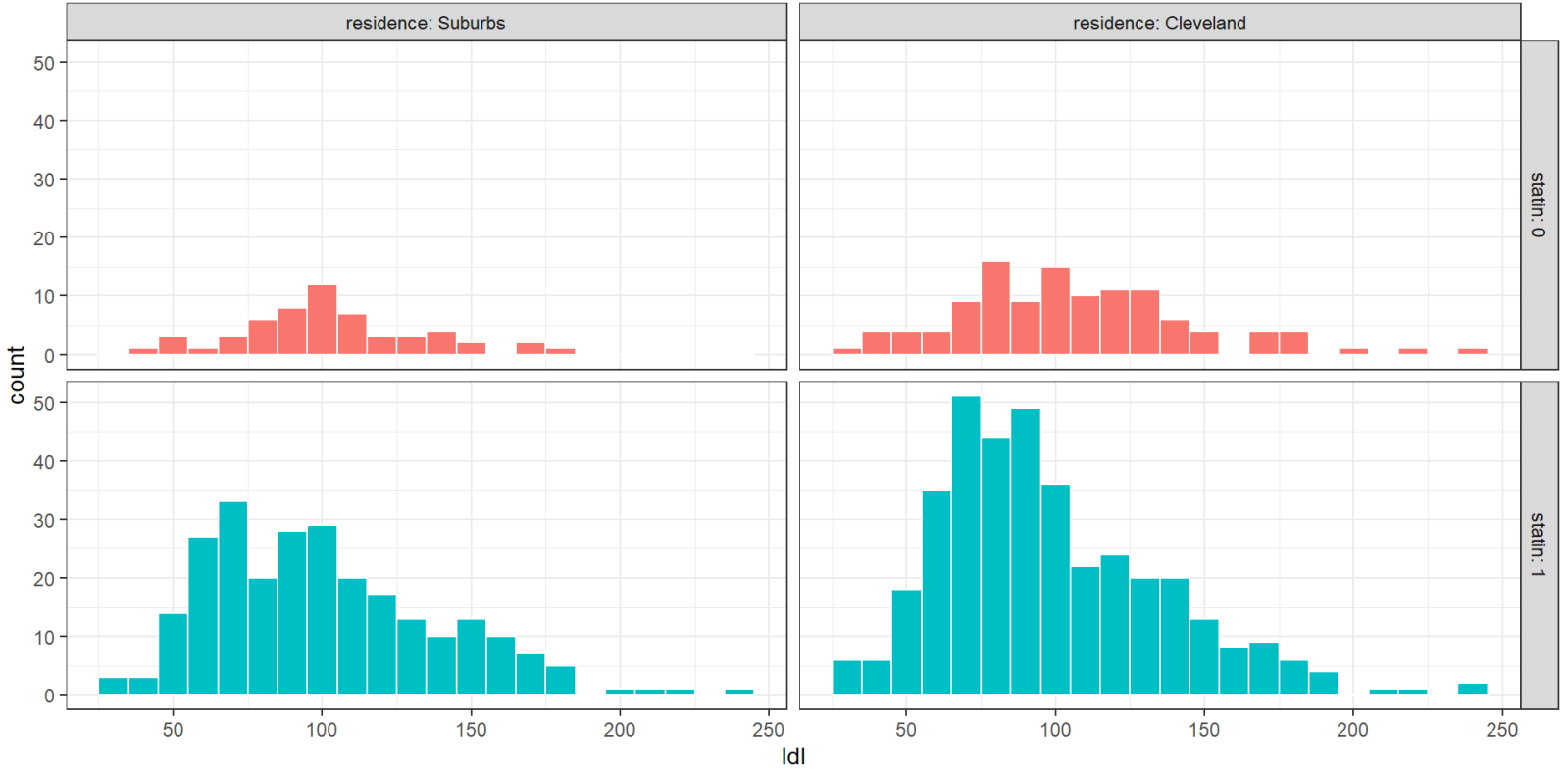
Using `facet_grid()`

```
1  tempdat <- dm1000 |> filter(complete.cases(ldl, statin))
2  ggplot(data = tempdat, aes(x = ldl, fill = factor(statin))) +
3    geom_histogram(binwidth = 10, col = "white") +
4    facet_grid(statin ~ ., labeller = "label_both") +
5    guides(fill = "none")
```



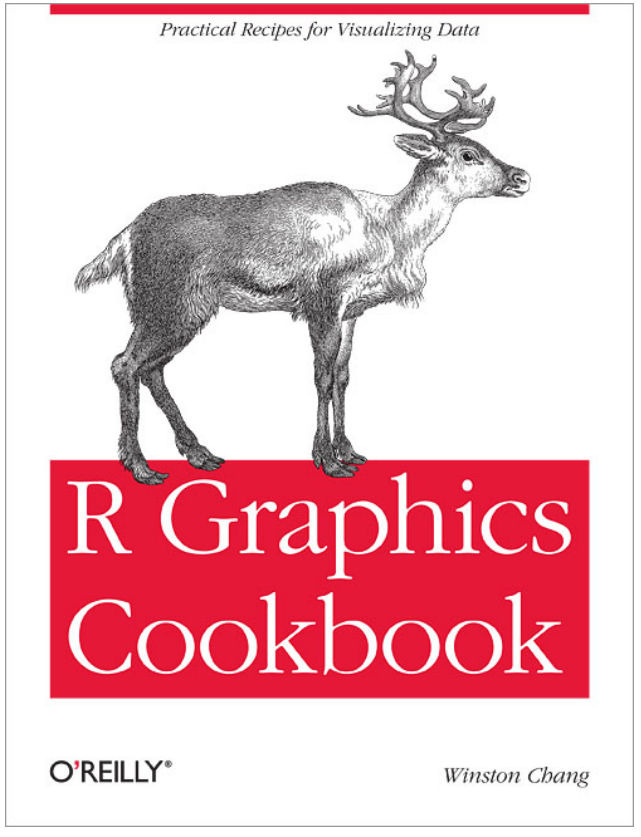
facet_grid(): two groupings

```
1  tempdat <- dm1000 |> filter(complete.cases(ldl, statin, residence))
2  ggplot(data = tempdat, aes(x = ldl, fill = factor(statin))) +
3    geom_histogram(binwidth = 10, col = "white") +
4    facet_grid(statin ~ residence, labeller = "label_both") +
5    guides(fill = "none")
```



My Main Source for `ggplot2` Visualization Recipes

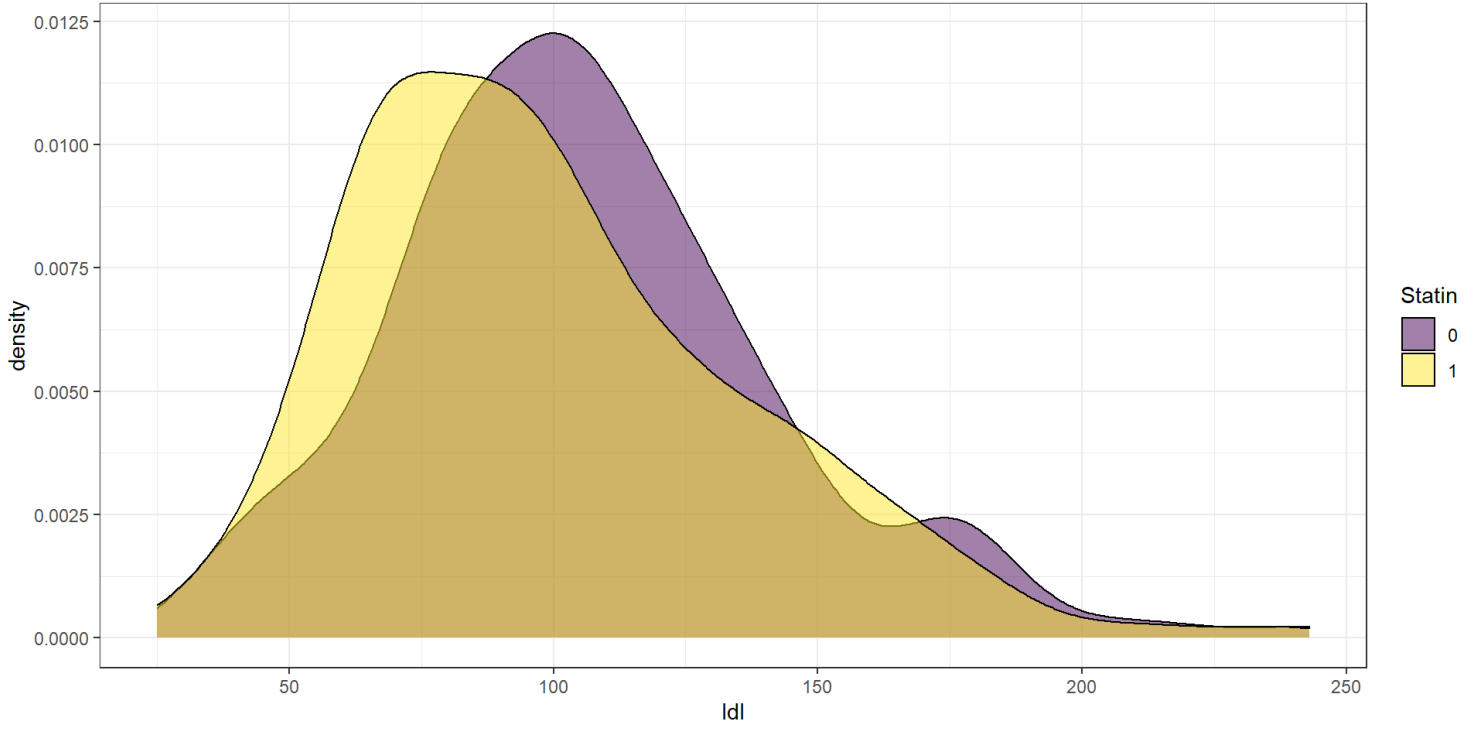
https://r-graphics.org/ (Second Edition)



Comparison of densities

This plot ignores the relative frequencies.

```
1  tempdat <- dm1000 |> filter(complete.cases(ldl, statin))
2  ggplot(data = tempdat, aes(x = ldl, fill = factor(statin))) +
3    geom_density(alpha = 0.5) + scale_fill_viridis_d() +
4    labs(fill = "Statin")
```



Numerical Summaries for Two Groups

```
dm1000 |> filter(complete.cases(statin, ldl)) |>
  group_by(statin) |>
  summarize(n = n(), min = min(ldl), med = median(ldl),
            max = max(ldl), mean = mean(ldl),
            sd = sd(ldl)) |>
  kbl(digits = 2)
```

statin	n	min	med	max	mean	sd
0	176	34	102	243	106.22	36.05
1	646	25	92	241	99.22	37.21

- Difference in mean(LDL) between the two samples?

Using `favstats` for LDL by Statin

```
1 mosaic::favstats(ldl ~ statin, data = dm1000)
```

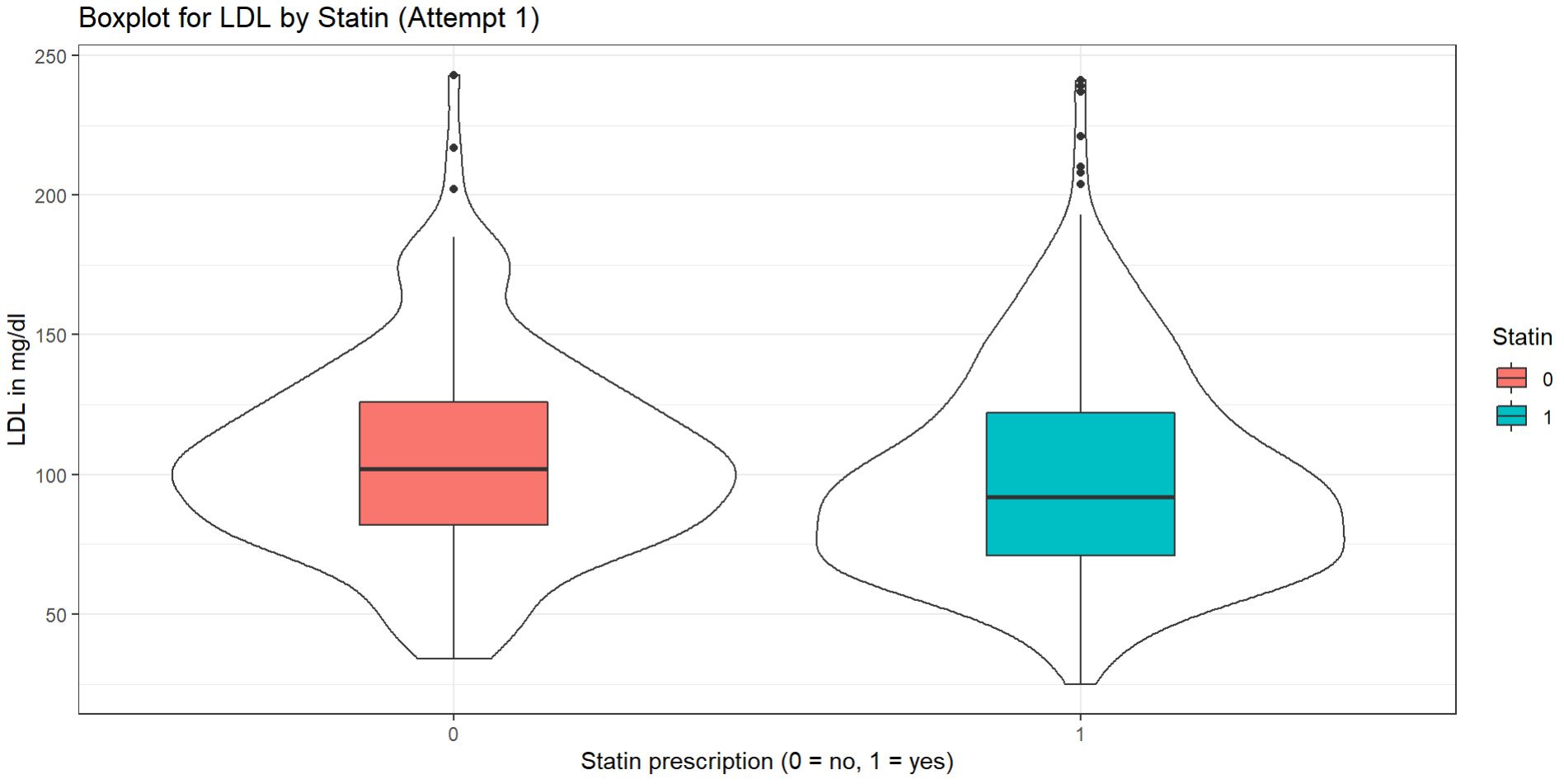
```
  statin min Q1 median  Q3 max      mean        sd   n missing
1      0  34 82    102 126 243 106.22159 36.04619 176      66
2      1  25 71     92 122 241  99.22136 37.20972 646     112
```

Comparison Boxplot (LDL by statin)

Attempt 1

```
tempdat <- dm1000 |> filter(complete.cases(ldl, statin))

ggplot(data = tempdat, aes(x = factor(statin), y = ldl)) +
  geom_violin() +
  geom_boxplot(aes(fill = factor(statin)), width = 0.3) +
  labs(x = "Statin prescription (0 = no, 1 = yes)",
       y = "LDL in mg/dl", fill = "Statin",
       title = "Boxplot for LDL by Statin (Attempt 1)")
```

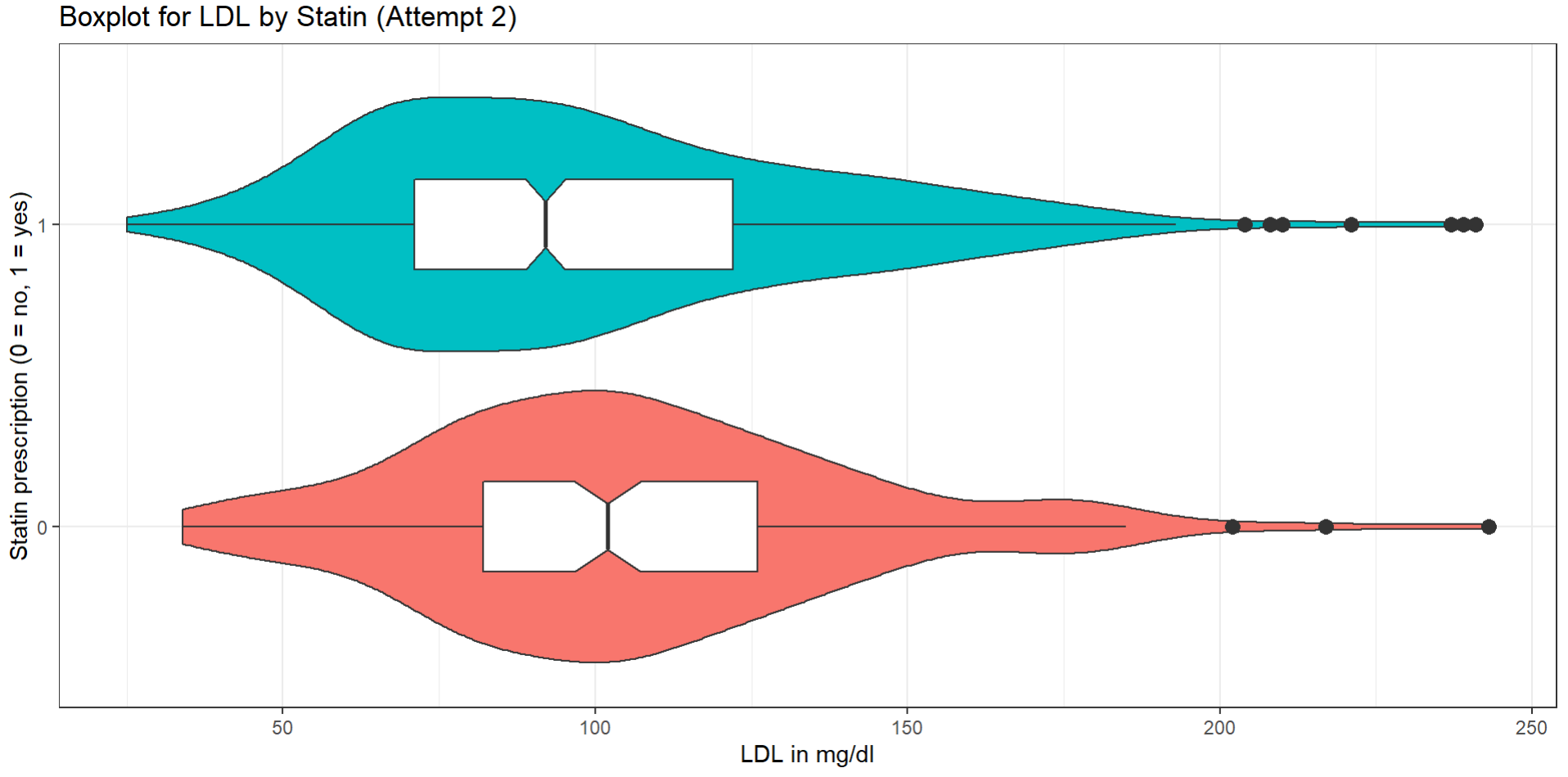
Comparison Boxplot (LDL by statin)



Try 2: Boxplot for LDL and statin

```
tempdat <- dm1000 |> filter(complete.cases(ldl, statin))

ggplot(data = tempdat, aes(x = factor(statin), y = ldl)) +
  geom_violin(aes(fill = factor(statin))) +
  geom_boxplot(width = 0.3, outlier.size = 3, notch = TRUE) +
  coord_flip() +
  guides(fill = "none") +
  labs(x = "Statin prescription (0 = no, 1 = yes)",
       y = "LDL in mg/dl",
       title = "Boxplot for LDL by Statin (Attempt 2)")
```

Try 2: Boxplot for LDL and statin



Setting Up Third Try

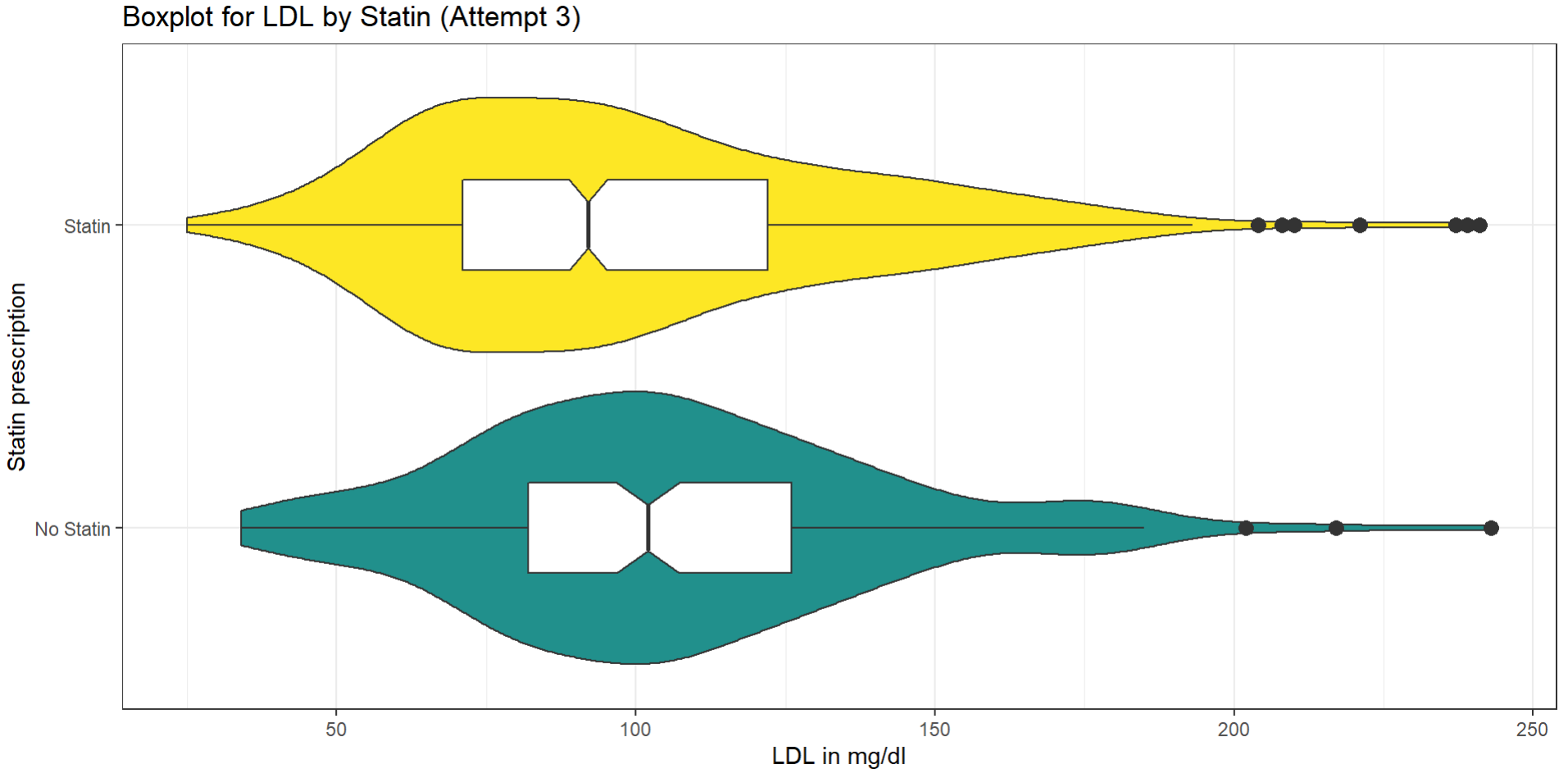
```
dm_for_boxplot <- dm1000 |>
  filter(complete.cases(statin, ldl)) |>
  mutate(statin_f = fct_recode(factor(statin),
                               "No Statin" = "0",
                               "Statin" = "1")) |>
  select(subject, ldl, statin_f, statin)

head(dm_for_boxplot, 3) # print first three rows
```

```
# A tibble: 3 × 4
  subject   ldl statin_f  statin
  <chr>   <dbl> <fct>      <dbl>
1 M-0001    221 Statin         1
2 M-0002    116 No Statin      0
3 M-0003     52 Statin         1
```

Third Try on Boxplot for LDL by Statin Use

```
ggplot(data = dm_for_boxplot, aes(x = statin_f, y = ldl)) +
  geom_violin(aes(fill = statin_f)) +
  geom_boxplot(width = 0.3, outlier.size = 3, notch = TRUE) +
  coord_flip() + guides(fill = "none") +
  scale_fill_viridis_d(begin = 0.5, option = "D") +
  labs(x = "Statin prescription",
       y = "LDL in mg/dl",
       title = "Boxplot for LDL by Statin (Attempt 3)")
```

Third Try on Boxplot for LDL by Statin Use



95% CI for difference in means

We want to estimate the difference between the population mean LDL WITH statin and population mean LDL WITHOUT statin.

The sample means, you'll remember, are:

```
mosaic::favstats(ldl ~ statin, data = dm1000) |>
  select(statin, n, mean, sd, missing) |>
  kbl(digits = 2) |> kable_styling(font_size = 24)
```

statin	n	mean	sd	missing
0	176	106.22	36.05	66
1	646	99.22	37.21	112

95% CI for difference in means

- **If** we are willing to assume that LDL follows a Normal distribution in each statin group, then we can use a linear model with one predictor.

```
m2 <- lm(ldl ~ statin, data = dm1000)
```

Coefficients of Model m2

```
1 m2
```

```
Call:
lm(formula = ldl ~ statin, data = dm1000)

Coefficients:
(Intercept)       statin
      106.2         -7.0
```

```
1 tidy(m2, conf.int = TRUE, conf.level = 0.95) |>
2   select(term, estimate, conf.low, conf.high)
```

```
# A tibble: 2 × 4
  term        estimate conf.low conf.high
  <chr>          <dbl>    <dbl>     <dbl>
1 (Intercept)   106.      101.     112.
2 statin         -7.00    -13.2     -0.831
```

Using `t.test` to get same result

```
mosaic::favstats(ldl ~ statin, data = dm1000) |>
  select(statin, n, mean, sd, missing) |>
  kbl(digits = 2) |> kable_styling(font_size = 24)
```

statin	n	mean	sd	missing
0	176	106.22	36.05	66
1	646	99.22	37.21	112

```
tt <- t.test(ldl ~ statin, data = dm1000,
        var.equal = TRUE, conf.level = 0.95)
tidy(tt) |> select(estimate, conf.low, conf.high) |>
  kbl(digits = 3) |> kable_styling(font_size = 32)
```

estimate	conf.low	conf.high
7	0.831	13.17

95% CI for difference between population means via the bootstrap

If we are not willing to assume a Normal distribution for LDL in either the statin or the "no statin" group, then we could use a bootstrap approach.

```
source("c08/data/Love-boost.R")

set.seed(123123)
bootdif(y = dm1000$ldl, g = factor(dm1000$statin), conf.level = 0.95)
```

```
Mean Difference           0.025           0.975
     -7.0002287    -13.3055407     -0.8186563
```

The `bootdif` function

from Love-boost.R

```
`bootdif` <-
  function(y, g, conf.level=0.95, B.reps = 2000) {
    lowq = (1 - conf.level)/2
    g <- as.factor(g)
    a <- attr(Hmisc::smean.cl.boot(y[g==levels(g)[1]],
                                   B=B.reps, reps=TRUE),'reps')
    b <- attr(Hmisc::smean.cl.boot(y[g==levels(g)[2]],
                                   B=B.reps, reps=TRUE),'reps')
    meandif <- diff(tapply(y, g, mean, na.rm=TRUE))
    a.b <- quantile(b-a, c(lowq,1-lowq))
    res <- c(meandif, a.b)
    names(res) <- c('Mean Difference',lowq, 1-lowq)
    res
  }
```

Assumptions behind our intervals

Assumptions these intervals share:

- random samples from the populations of interest
- independent samples (samples aren't paired or matched)

Additional assumptions for linear model:

- Normal distribution in each group (statin and "no statin")
- variance in each group (statin and "no statin") is equal

95% confidence intervals for mean LDL

95% CIs for LDL $\mu_{NoStatin} - \mu_{Statin}$

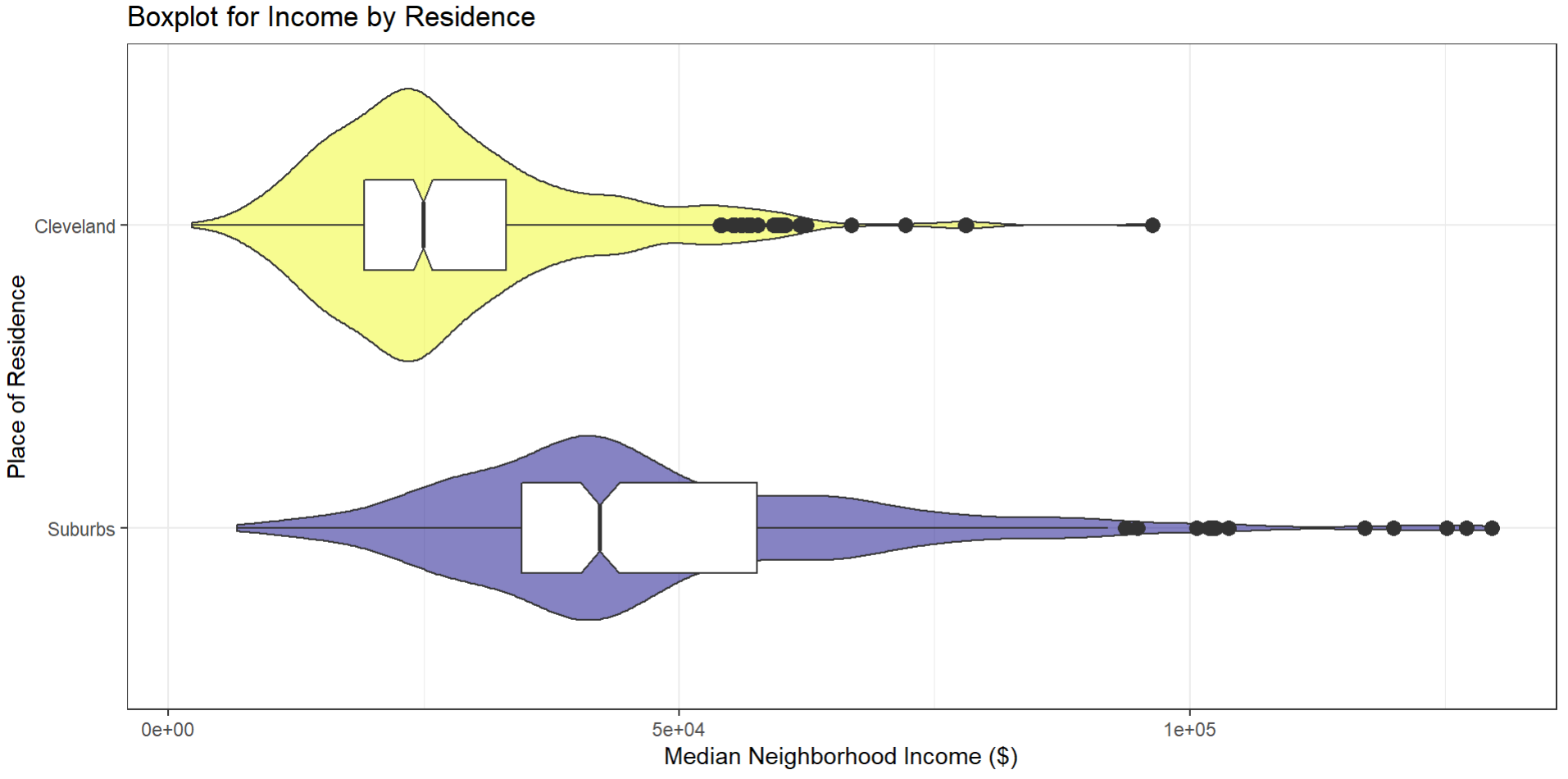
Approach	Estimate	95% CI
linear model	7.00	(0.83, 13.17)
bootstrap	7.00	(0.82, 13.31)

Are our conclusions meaningfully different if we do (or do not) assume Normal population distributions of LDL within each group (statin and no statin)?

Comparing Income by Residence

```
tempdat <- dm1000 |> filter(complete.cases(n_income, residence))

ggplot(data = tempdat, aes(x = residence, y = n_income)) +
  geom_violin(aes(fill = residence)) +
  geom_boxplot(width = 0.3, outlier.size = 3, notch = TRUE) +
  coord_flip() + guides(fill = "none") +
  scale_fill_viridis_d(alpha = 0.5, option = "C") +
  labs(x = "Place of Residence",
       y = "Median Neighborhood Income ($)",
       title = "Boxplot for Income by Residence")
```

Comparing Income by Residence



Sample Means, Medians

```
mosaic::favstats(n_income ~ residence, data = dm1000) |>
  select(residence, n, mean, median, sd, missing) |>
  kbl(digits = 0) |> kable_styling(font_size = 32)
```

residence	n	mean	median	sd	missing
Suburbs	371	47251	42223	20333	0
Cleveland	601	27725	24907	13031	0

Estimating 90% CI for Difference in Means

Linear Model for Income by Residence

```
m3 <- lm(n_income ~ residence, data = dm1000)
tidy(m3, conf.int = TRUE, conf.level = 0.90) |>
  select(term, estimate, conf.low, conf.high)
```

```
# A tibble: 2 × 4
  term                estimate conf.low conf.high
  <chr>                  <dbl>    <dbl>     <dbl>
1 (Intercept)           47251.   45866.    48637.
2 residenceCleveland   -19527.  -21289.   -17765.
```

Bootstrap Approach

```
set.seed(443322)
bootdif(y = dm1000$n_income, g = dm1000$residence,
        conf.level = 0.90)
```

```
Mean Difference            0.05            0.95
      -19526.68       -21468.71       -17545.30
```

90% CI for difference in mean Income

90% CIs for Income $\mu_{Suburbs} - \mu_{Cleveland}$

Approach	Estimate	90% CI
linear model	19527	(17765, 21289)
bootstrap	19527	(17545, 21469)

Are our conclusions meaningfully different if we do (or do not) assume Normal population distributions of neighborhood income within each residence group?

Save the `dm1000` tibble as an R data set

- Preserves all changes in your R work (factors, etc.)

```
write_rds(dm1000, "c08/data/dm_1000.Rds")
```

We'll load this in when we work with these data in Class 09.

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
```