431 Class 08

thomaselove.github.io/431

2021-09-16

Today's Agenda

- Building Visualizations to Compare Two Distributions
  - Confidence Intervals for a Difference between Means
- Building Visualizations to Compare $>$ 2 Distributions

Today's R Packages

```
library(broom) # for tidying up output
library(janitor)
library(knitr)
library(magrittr)
library(naniar)
library(patchwork)
library(readxl) # new today, read in .xls or .xlsx files
library(tidyverse)

theme_set(theme_bw())
```

Today's Data

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

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

- There are also functions called `read_xls()` and `read_xlsx()` available in the `readxl` package.

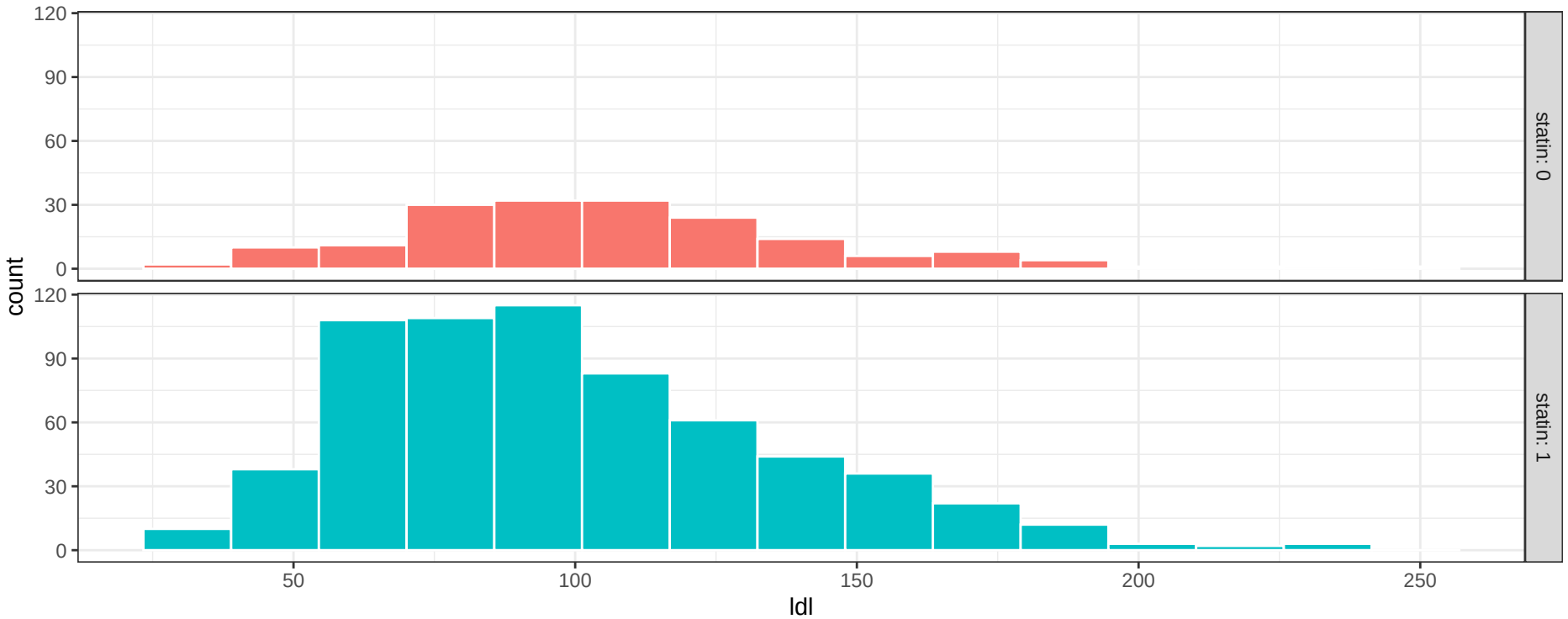
The dm1000 tibble

```
# A tibble: 1,000 x 17
   subject   age insurance  n_income    ht    wt   sbp
   <chr>   <dbl> <fct>         <dbl> <dbl> <dbl> <dbl>
 1 M-0001     55 Medicaid      29853  1.63 103.    145
 2 M-0002     52 Commercial    31248  1.75 112.    151
 3 M-0003     69 Medicare      23362  1.65  74.9   127
 4 M-0004     57 Medicaid      26033  1.63  81.4   125
 5 M-0005     68 Medicare      85374  1.69  92.6   120
 6 M-0006     56 Medicaid      31273  1.71  54.6   127
 7 M-0007     54 Commercial    25445  1.68  81.6   114
 8 M-0008     45 Medicare      67526  1.69  80.6   166
 9 M-0009     61 Medicare      15203  1.91  86.7   111
10 M-0010     63 Medicaid      17628  1.86 123.    146
# ... with 990 more rows, and 10 more variables:
#   dbp <dbl>, a1c <dbl>, ldl <dbl>, tobacco <fct>,
#   statin <dbl>, eye_exam <dbl>,
#   race_ethnicity <fct>, sex <fct>, county <fct>,
```

Comparing Two Distributions

LDL cholesterol and statin prescription?

```
dm1000 %>% filter(complete.cases(ldl, statin)) %>%
  ggplot(data = ., aes(x = ldl, fill = factor(statin))) +
  geom_histogram(bins = 15, col = "white") +
  facet_grid(statin ~ ., labeller = "label_both") +
  guides(fill = "none")
```



Faceting

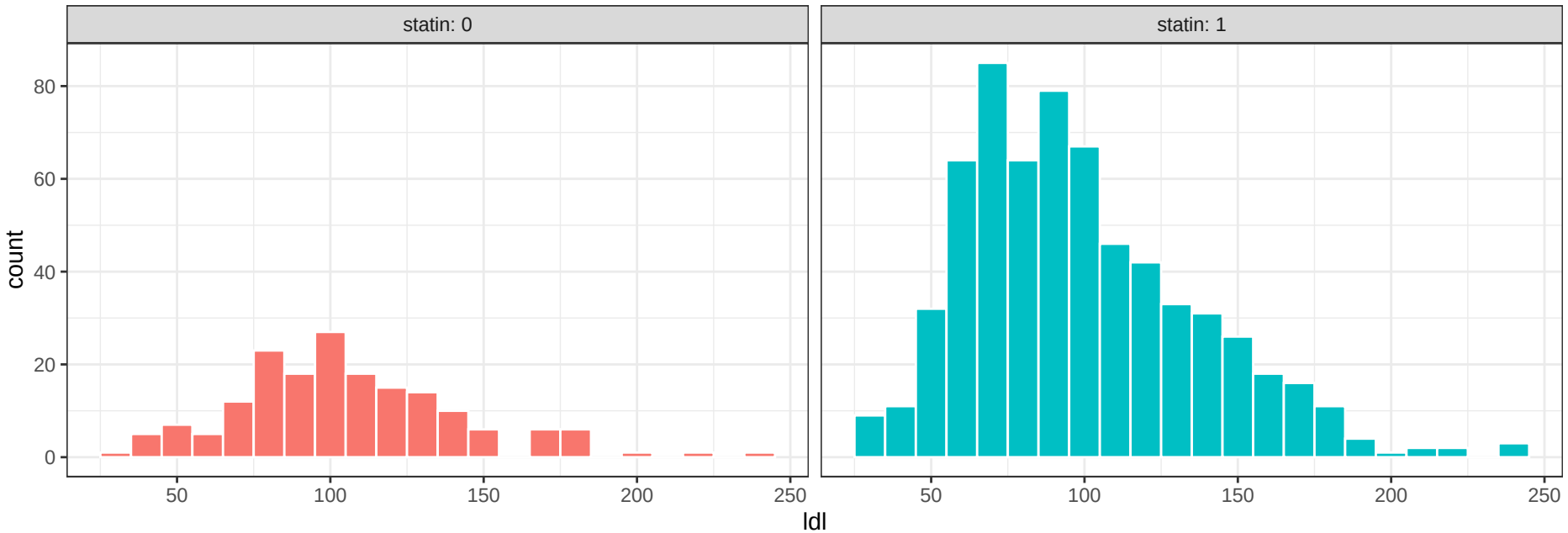
It's very useful to split data into groups and plot each group separately to make comparisons across the groups. We can then draw those subplots side by side.

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.

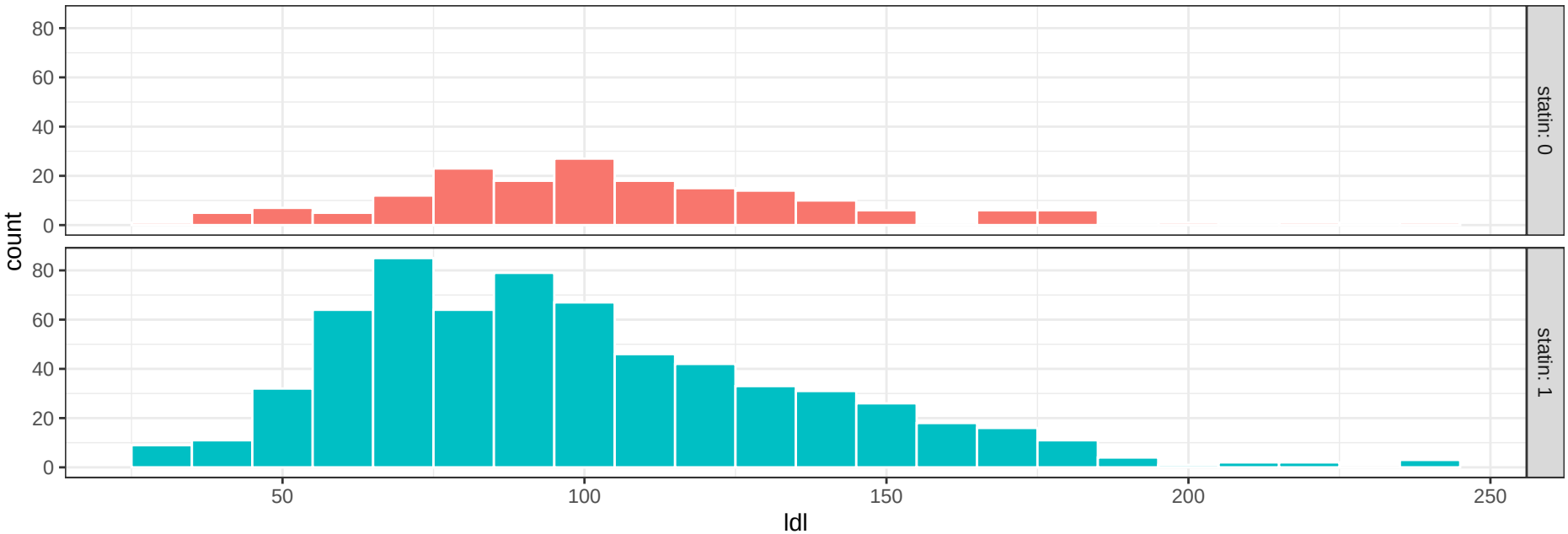
Using `facet_wrap()`

```
dm1000 %>% filter(complete.cases(ldl, statin)) %>%
  ggplot(data = ., aes(x = ldl, fill = factor(statin))) +
  geom_histogram(binwidth = 10, col = "white") +
  facet_wrap(~ statin, labeller = "label_both") +
  guides(fill = "none")
```



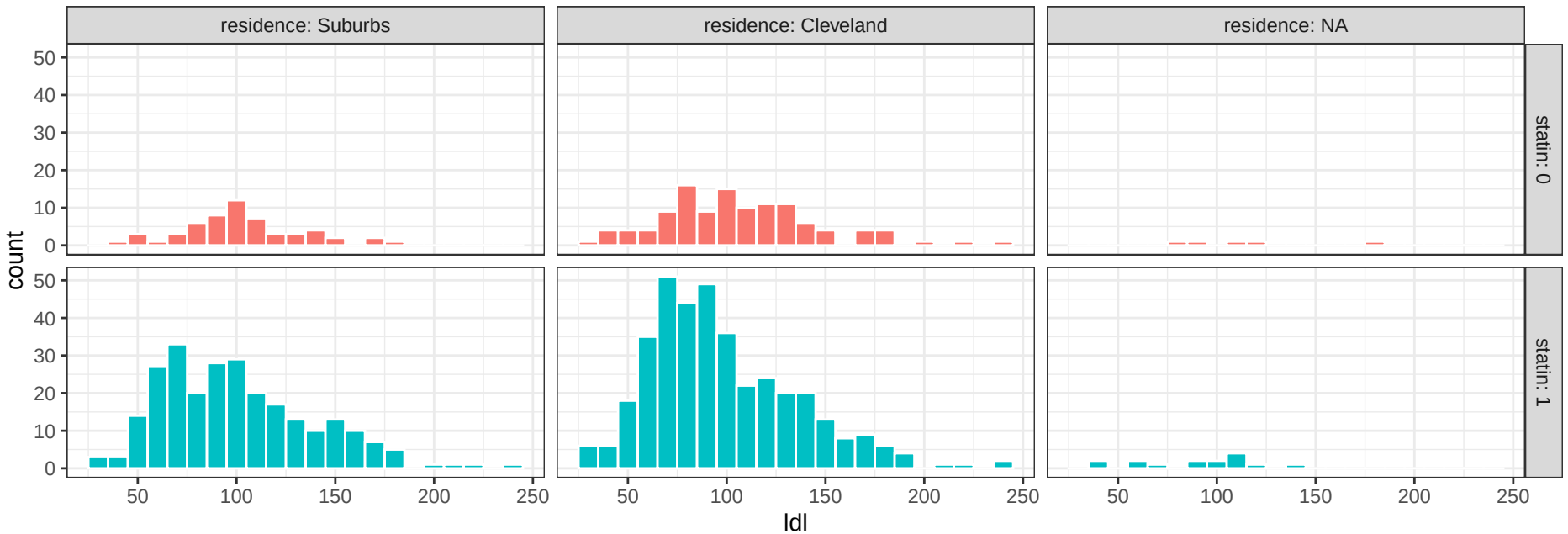
Using facet_grid()

```
dm1000 %>% filter(complete.cases(ldl, statin)) %>%
  ggplot(data = ., aes(x = ldl, fill = factor(statin))) +
  geom_histogram(binwidth = 10, col = "white") +
  facet_grid(statin ~ ., labeller = "label_both") +
  guides(fill = "none")
```



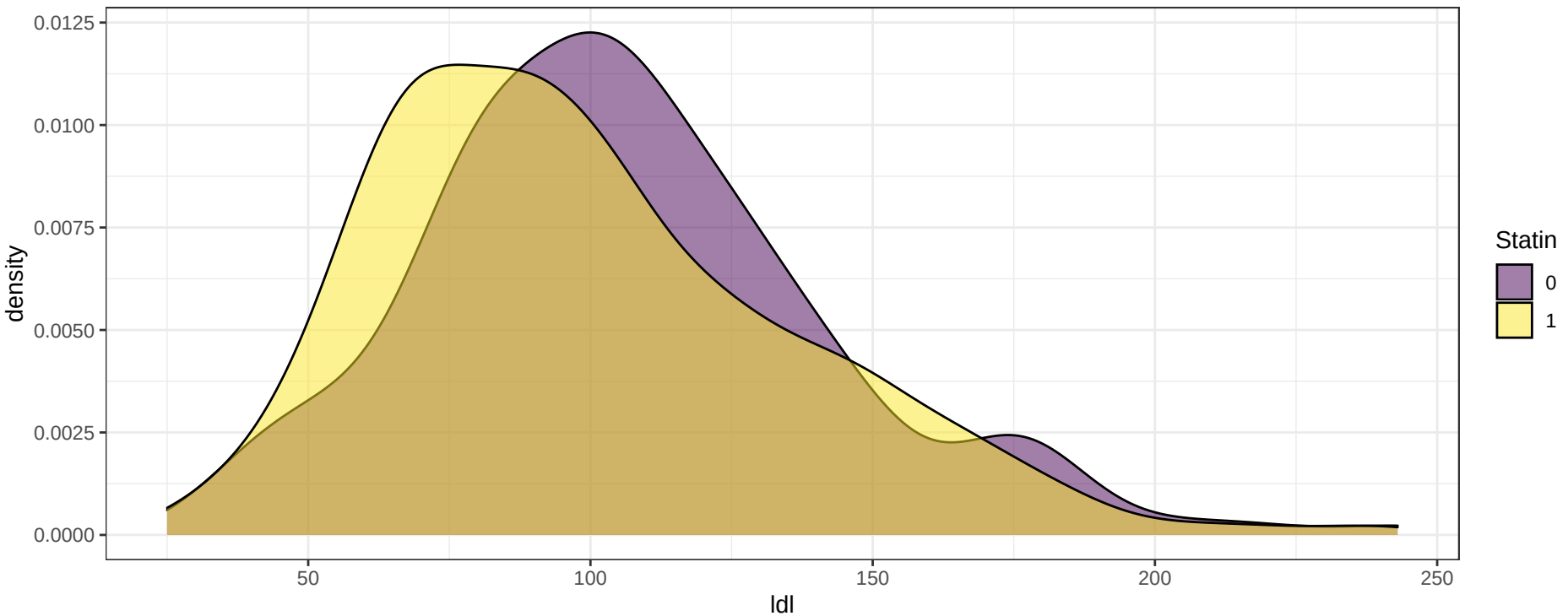
Using `facet_grid()` with two groupings

```
dm1000 %>% filter(complete.cases(ldl, statin)) %>%
  ggplot(data = ., aes(x = ldl, fill = factor(statin))) +
  geom_histogram(binwidth = 10, col = "white") +
  facet_grid(statin ~ residence, labeller = "label_both") +
  guides(fill = "none")
```



Comparison of densities (ignores relative frequency)

```
dm1000 %>% filter(complete.cases(ldl, statin)) %>%
  ggplot(data = ., aes(x = ldl, fill = factor(statin))) +
  geom_density(alpha = 0.5) +
  scale_fill_viridis_d() +
  labs(fill = "Statin")
```



Numerical Summaries comparing 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)) %>%
  kable(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

- What is the difference in mean(LDL) between the two samples?

Using `favstats` to compare LDL by Statin group

```
dm1000 %$%
  mosaic::favstats(ldl ~ statin)
```

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

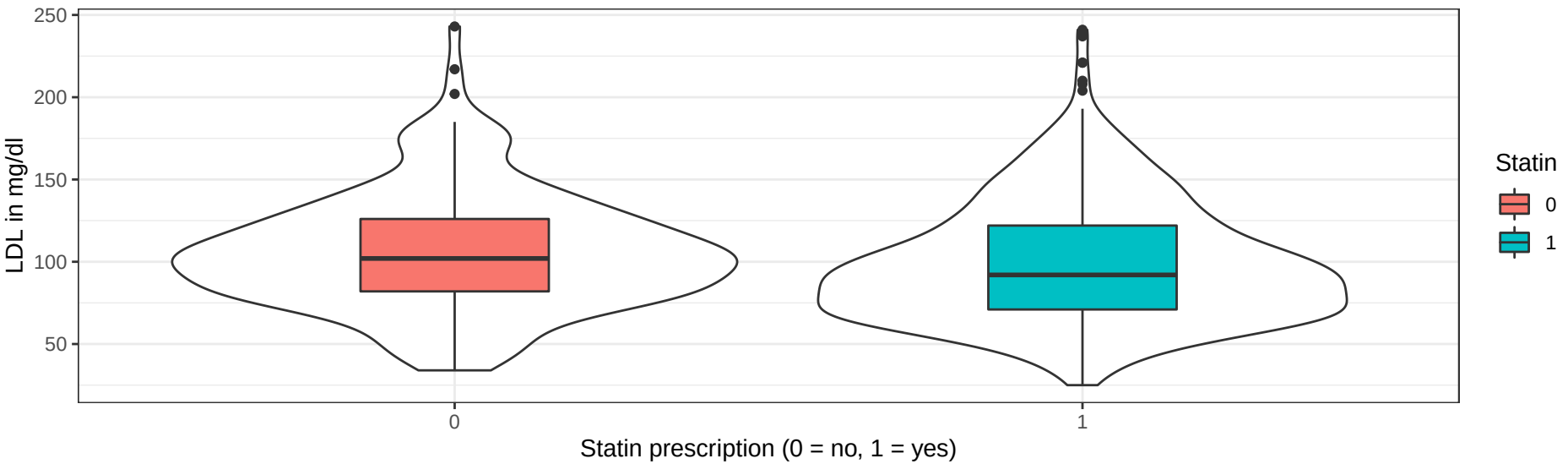
We would have obtained the same result with:

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

Comparison Boxplot with Violins (LDL and statin)

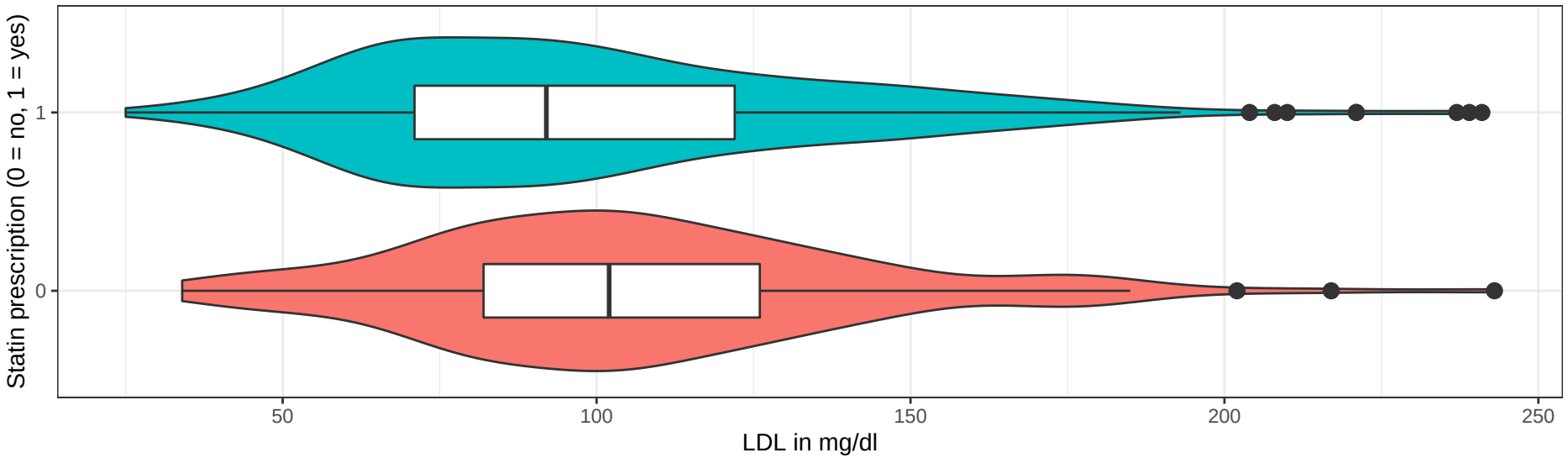
Here's a first attempt...

```
dm1000 %>% filter(complete.cases(ldl, statin)) %>%
  ggplot(data = ., 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")
```



Try 2: Boxplot with Violins for LDL and statin

```
dm1000 %>% filter(complete.cases(ldl, statin)) %>%
  ggplot(data = ., aes(x = factor(statin), y = ldl)) +
  geom_violin(aes(fill = factor(statin))) +
  geom_boxplot(width = 0.3, outlier.size = 3) +
  coord_flip() +
  guides(fill = "none") +
  labs(x = "Statin prescription (0 = no, 1 = yes)",
       y = "LDL in mg/dl")
```



Setting Up Third Try

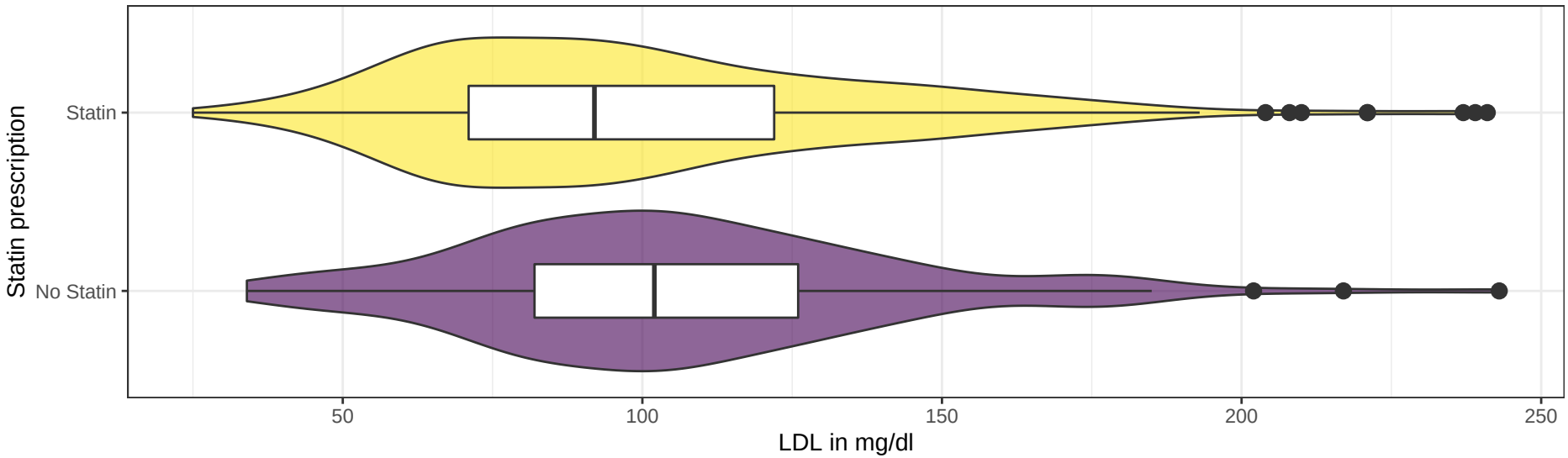
```
dm_for_boxplot <- dm1000 %>%
  filter(complete.cases(statin, ldl)) %>%
  mutate(statin_f = fct_recode(factor(statin),
                               "No Statin" = "0",
                               "Statin" = "1")) %>%
  select(subject, ldl, statin_f, statin)

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

```
# A tibble: 3 x 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) +
  coord_flip() + guides(fill = "none") +
  scale_fill_viridis_d(alpha = 0.6) +
  labs(x = "Statin prescription",
       y = "LDL in mg/dl")
```



95% confidence interval for difference between population mean LDL WITH statin and population mean LDL WITHOUT statin

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

```
m2 <- lm(ldl ~ statin, data = dm1000)
tidy(m2, conf.int = TRUE, conf.level = 0.95) %>%
  select(term, estimate, conf.low, conf.high) %>%
  kable(digits = 3)
```

term	estimate	conf.low	conf.high
(Intercept)	106.222	100.752	111.691
statin	-7.000	-13.170	-0.831

Alternative Approach to get same result

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

estimate	conf.low	conf.high
7	0.831	13.17

95% confidence interval for difference between population mean LDL WITH statin and population mean LDL WITHOUT statin

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("data/Love-boost.R")
set.seed(20210914)
dm1000 %$% bootdif(y = ldl,
                   g = factor(statin),
                   conf.level = 0.95)
```

```
Mean Difference            0.025            0.975
     -7.0002287     -12.9555578      -0.9043361
```

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
  }
```

95% confidence intervals for $\mu_{NoStatin} - \mu_{Statin}$

Approach	Estimate	95% CI	Normality Assumption?
linear model	7.00	(0.83, 13.17)	Yes
bootstrap	7.00	(0.90, 12.96)	No

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

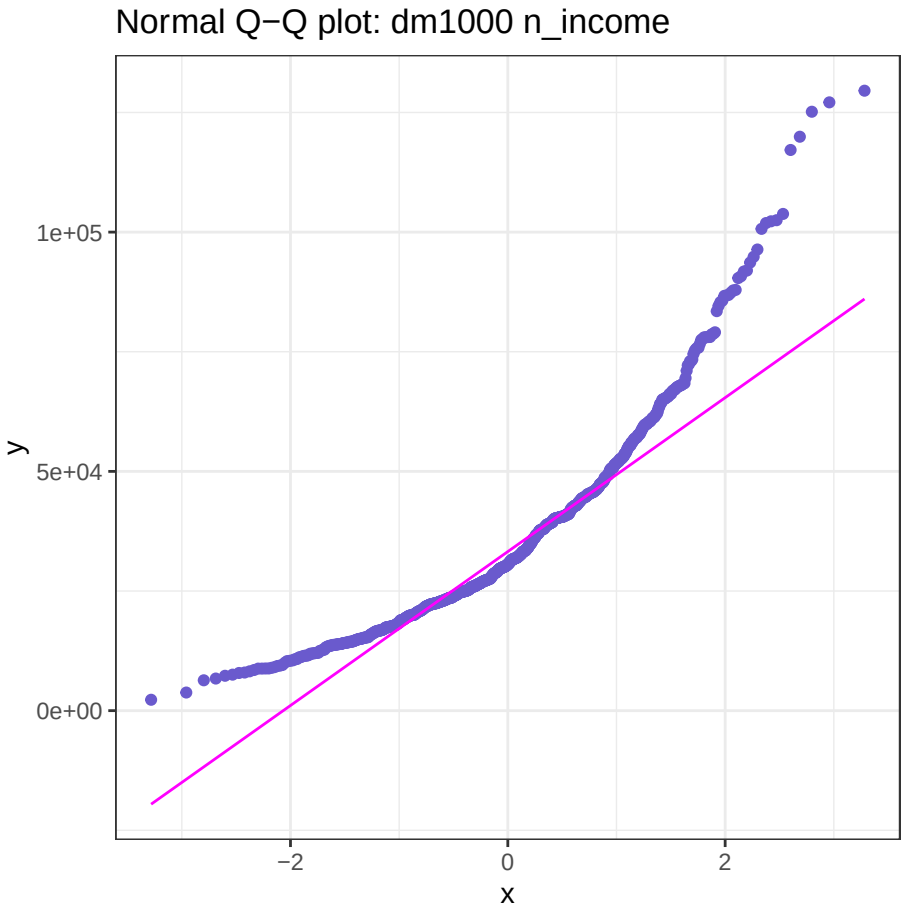
Comparing Multiple (more than 2) Batches of Data

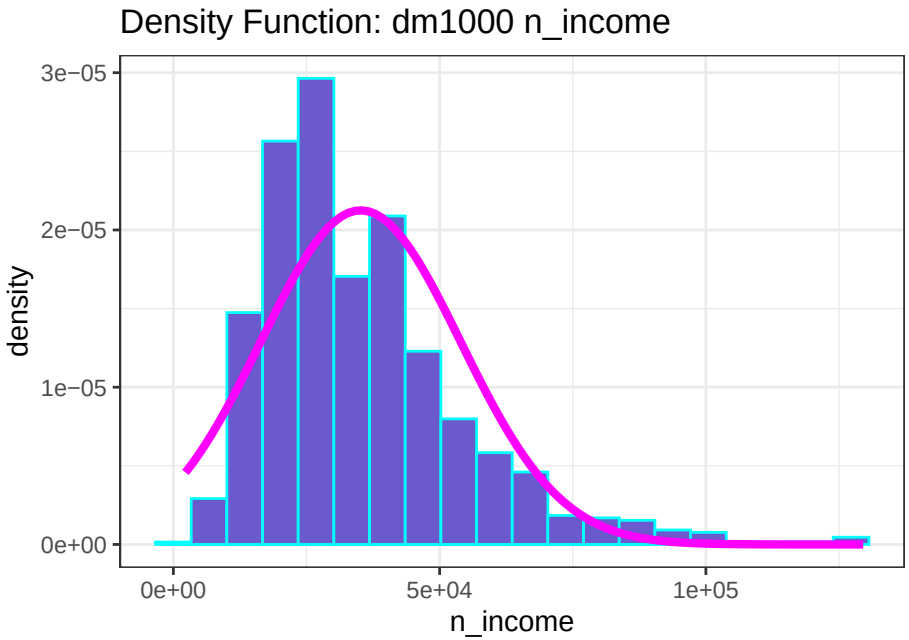
Stratify the subjects by primary `insurance`?

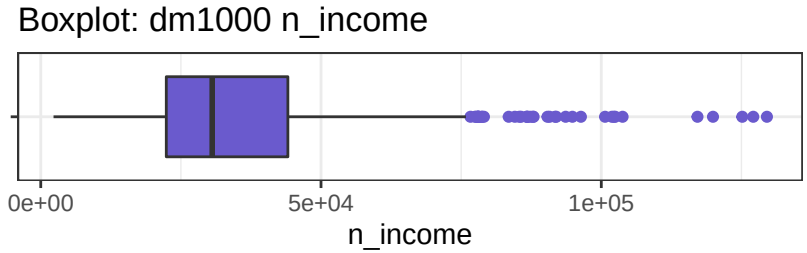
```
dm1000 %>% count(insurance) %>%
  mutate(pct = 100*n/sum(n)) %>%
  kable(digits = 1)
```

insurance	n	pct
Medicaid	330	33.0
Commercial	196	19.6
Medicare	432	43.2
Uninsured	42	4.2

Let's look at n_income





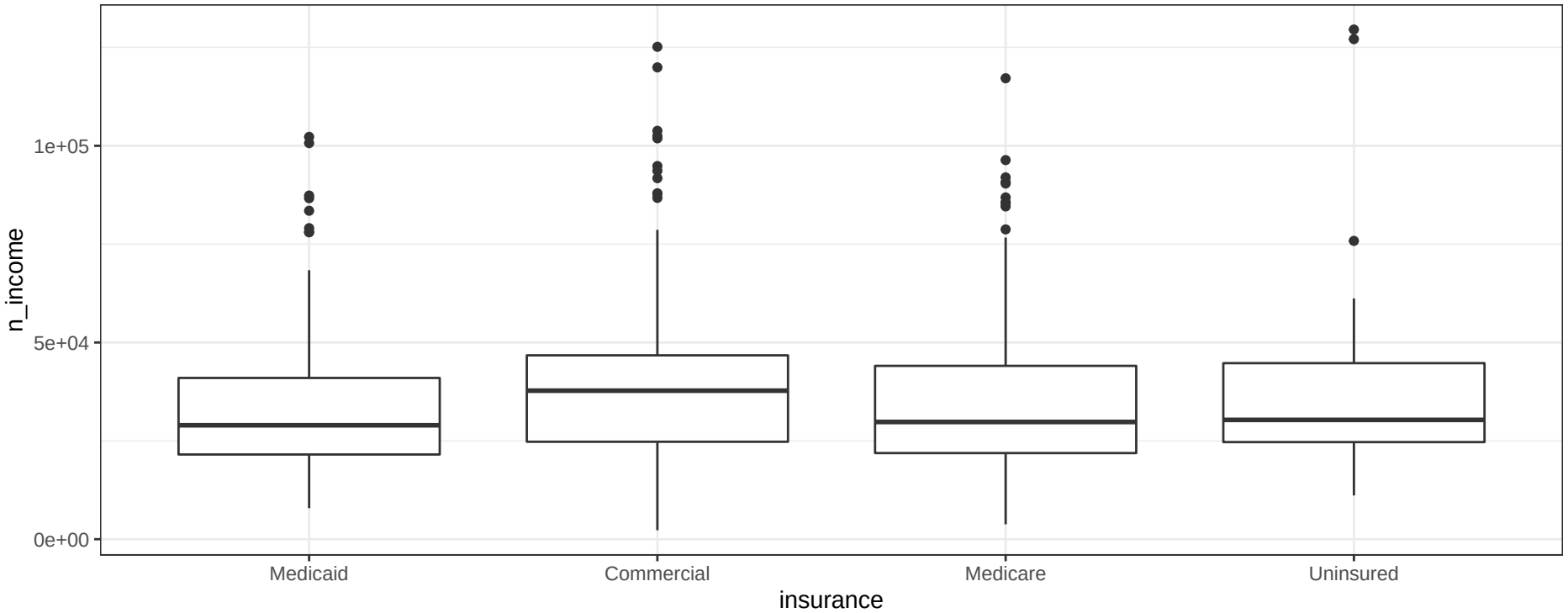


min	Q1	median	Q3	max	mean	sd	n	missing
2279	22393	30586	44085	129549	35178	18776	972	28

Compare n_income by insurance: Boxplot?

```
ggplot(dm1000, aes(x = insurance, y = n_income)) +
  geom_boxplot()
```

```
Warning: Removed 28 rows containing non-finite values
(stat_boxplot).
```



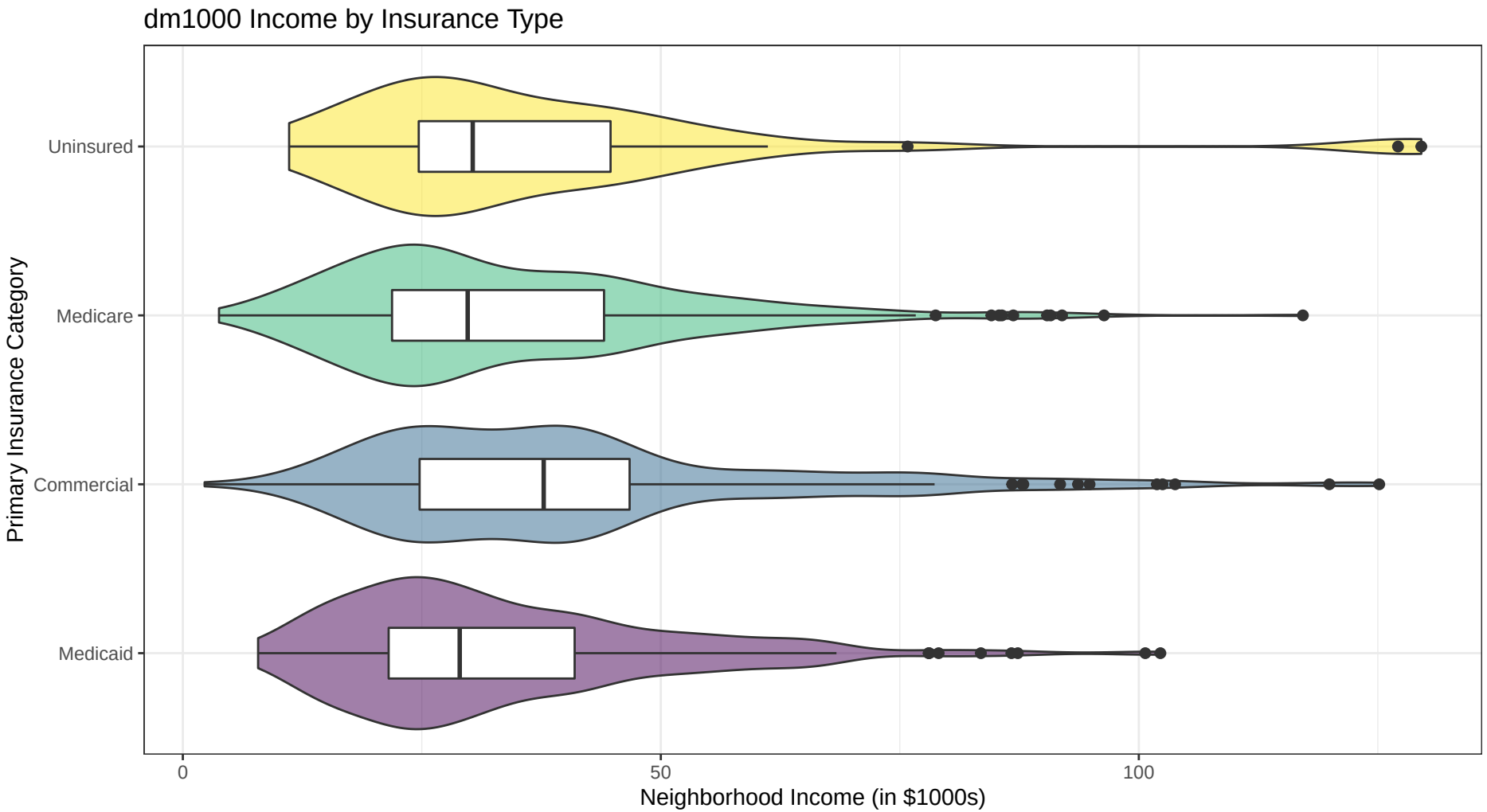
Build a better boxplot?

What am I doing in each line of this code?

```
dm1000 %>% filter(complete.cases(insurance, n_income)) %>%
  ggplot(data = ., aes(x = insurance, y = n_income/1000)) +
  geom_violin(aes(fill = insurance)) +
  geom_boxplot(width = 0.3, outlier.size = 2) +
  guides(fill = "none") +
  coord_flip() +
  scale_fill_viridis_d(alpha = 0.5) +
  labs(y = "Neighborhood Income (in $1000s)",
       x = "Primary Insurance Category",
       title = "dm1000 Income by Insurance Type")
```

- Result on next slide. . .

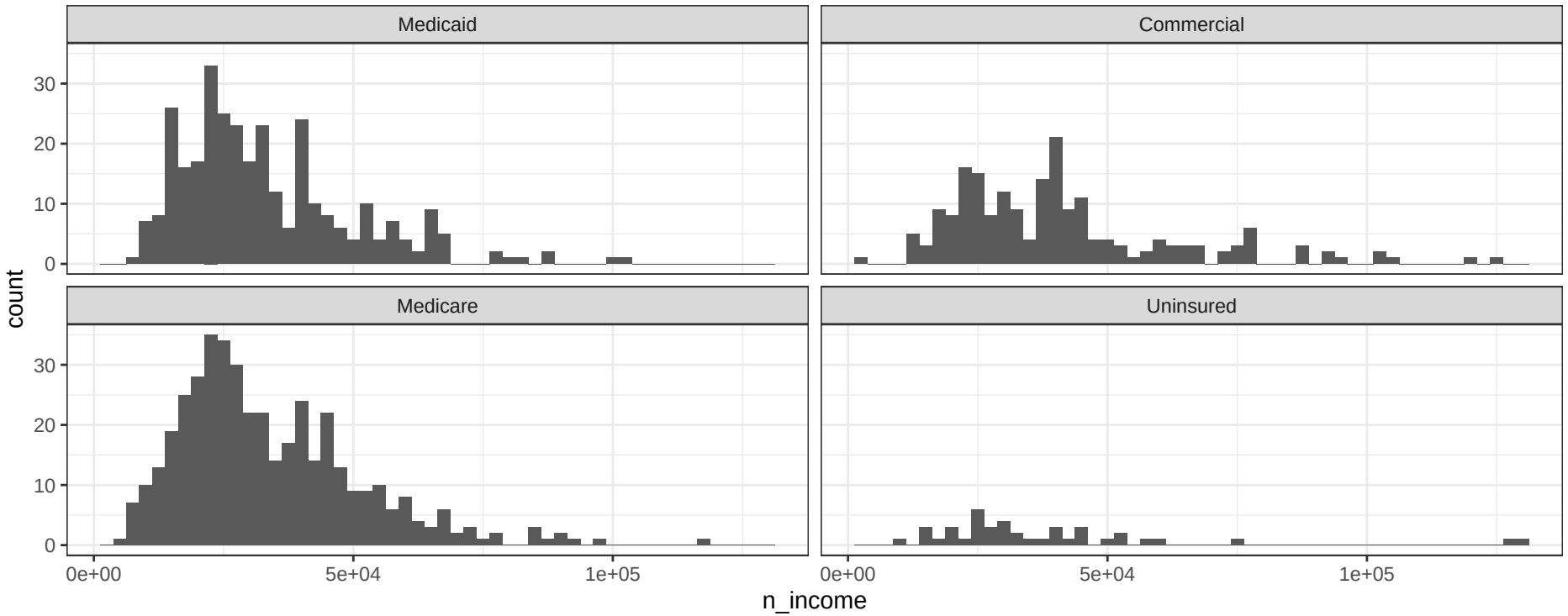
How does n_income vary by insurance in dm1000?



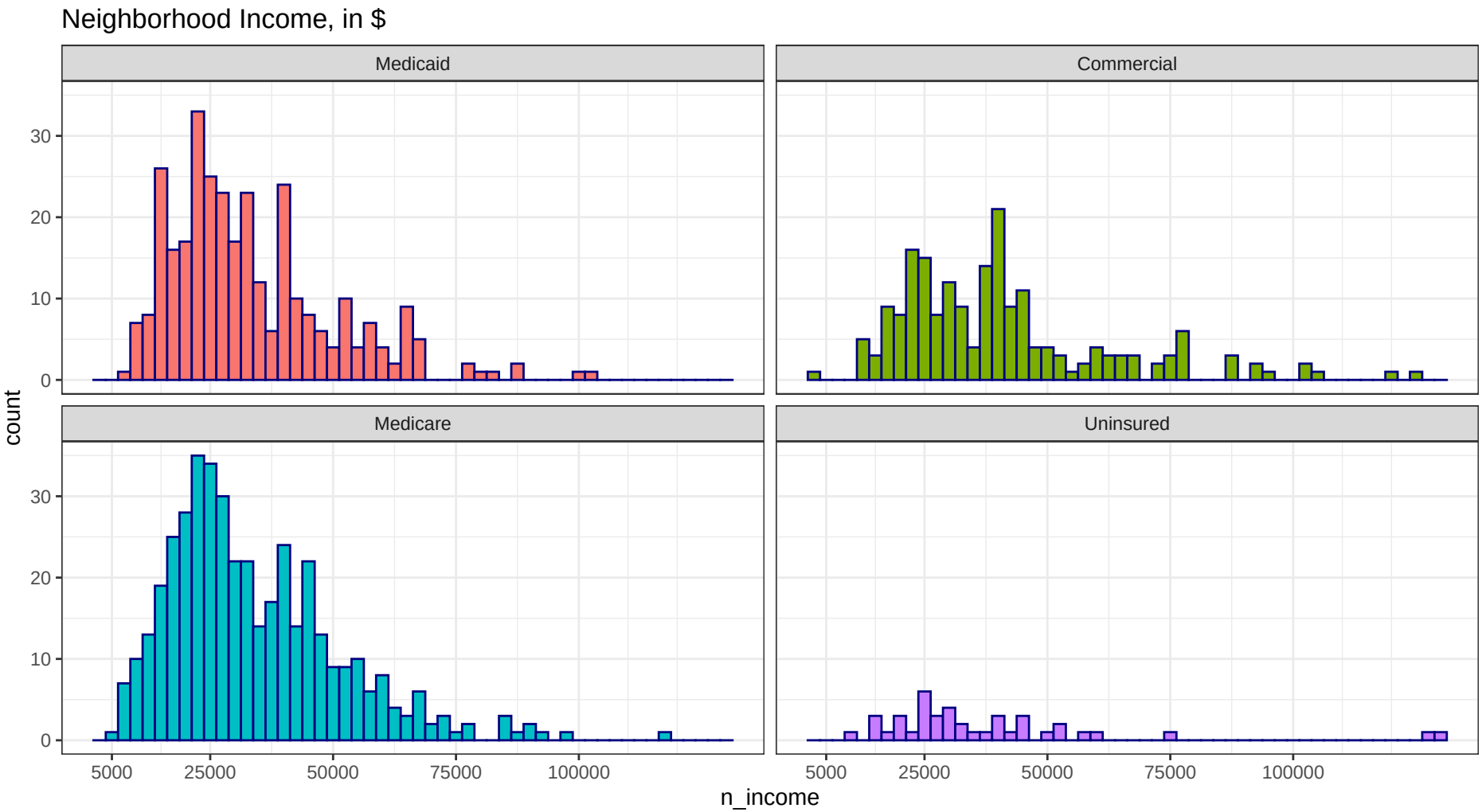
Faceted Histograms of n_income by insurance

```
ggplot(data = dm1000, aes(x = n_income)) +
  geom_histogram(binwidth = 2500) +
  facet_wrap(~ insurance)
```

```
Warning: Removed 28 rows containing non-finite values
(stat_bin).
```



Improving the Histograms (result)

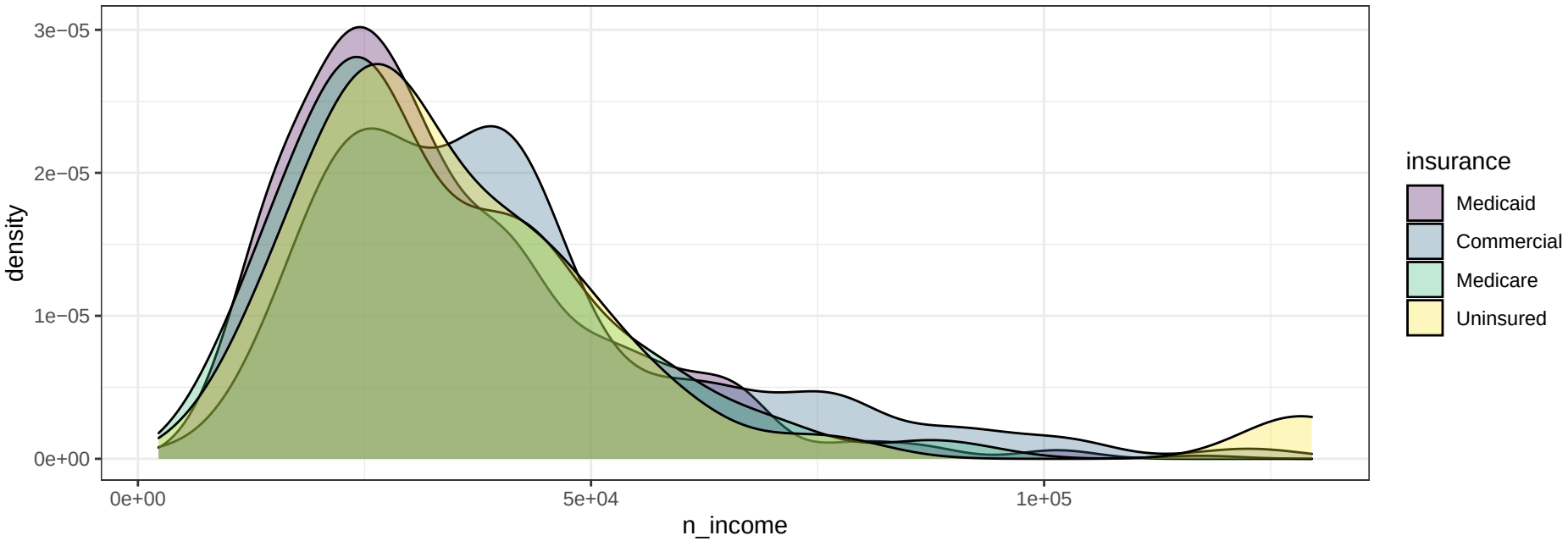


Improving the Histograms (code)

```
dm1000 %>% filter(complete.cases(n_income, insurance)) %>%
ggplot(data = ., aes(x = n_income, fill = insurance)) +
  geom_histogram(binwidth = 2500, col = "navy") +
  scale_x_continuous(
    breaks = c(5000, 25000, 50000, 75000, 100000)) +
  guides(fill = "none") +
  facet_wrap(~ insurance) +
  labs(title = "Neighborhood Income, in $")
```

Comparing Densities of n_income by insurance

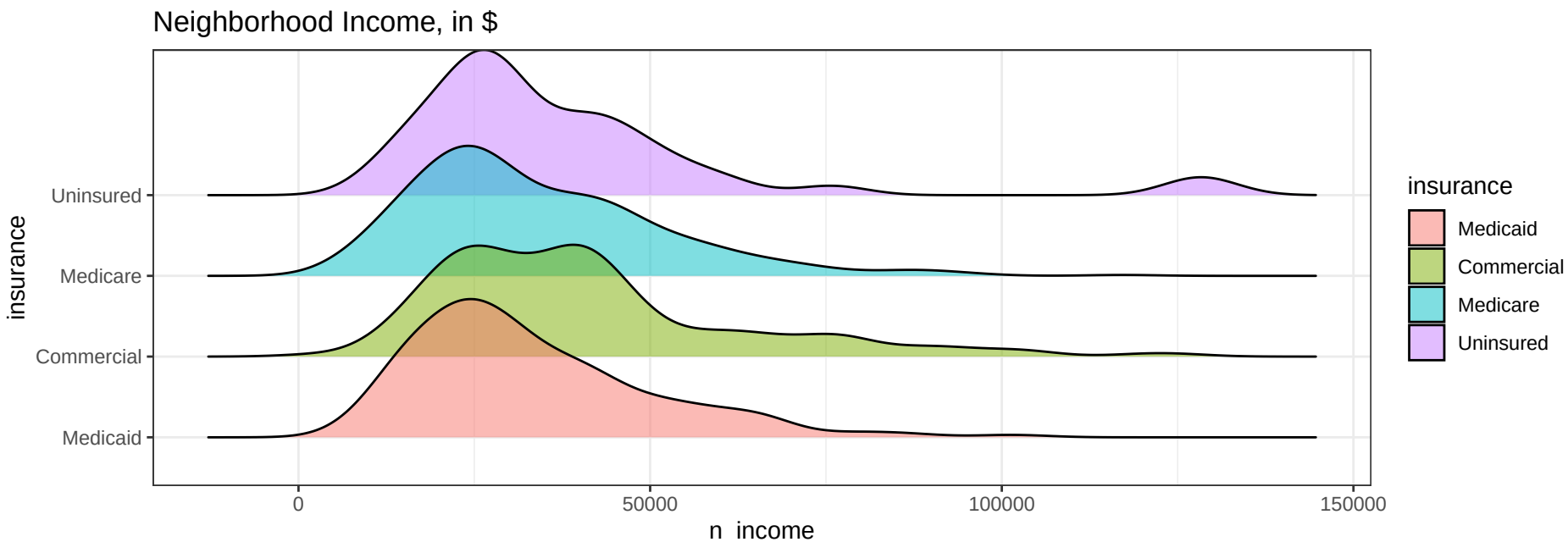
```
dm1000 %>% filter(complete.cases(n_income, insurance)) %>%
  ggplot(data = ., aes(x = n_income, fill = insurance)) +
  geom_density() +
  scale_fill_viridis_d(alpha = 0.3)
```



Using a Ridgeline Plot to Compare Densities

```
dm1000 %>% filter(complete.cases(n_income, insurance)) %>%
  ggplot(data = ., aes(x = n_income, y = insurance,
                       fill = insurance)) +
  geom_density_ridges(alpha = 0.5) +
  labs(title = "Neighborhood Income, in $")
```

```
Picking joint bandwidth of 5030
```



Sample Summaries of n_income

```
dm1000 %$% mosaic::favstats(n_income ~ insurance)
```

```
   insurance   min      Q1  median       Q3    max
1   Medicaid  7876 21528.5 28965.0 40986.00 102258
2 Commercial  2279 24767.0 37748.0 46736.00 125150
3   Medicare  3787 21883.5 29797.5 44062.75 117161
4  Uninsured 11121 24678.0 30328.0 44743.00 129549
      mean        sd   n missing
1 33150.74 16814.04 315      15
2 40715.02 21713.97 194       2
3 33883.50 17529.58 422      10
4 37874.73 24962.68  41       1
```

Comparing the Means of n_income

```
m1 <- lm(n_income ~ insurance, data = dm1000)

tidy(m1) %>% select(term, estimate) %>% kable(digits = 2)
```

term	estimate
(Intercept)	33150.74
insuranceCommercial	7564.28
insuranceMedicare	732.76
insuranceUninsured	4724.00

- What is m1's estimated n_income for each of the insurance groups?

m1: Estimated n_income for each insurance

```
n_income = 33150.74 + 7564.28 Commercial
                    + 732.76 Medicare + 4724.00 Uninsured
```

Insurance	Estimated n_income
Commercial	33150.74 + 7564.28 = 40715.02
Medicare	33150.74 + 732.76 = 33883.50
Uninsured	33150.74 + 4724.00 = 37874.74
Medicaid	33150.74

m1: Estimated n_income for each insurance

```
n_income = 33150.74 + 7564.28 Commercial
                    + 732.76 Medicare + 4724.00 Uninsured
```

Insurance	m1 est.	Sample mean(n_income)
Commercial	40715.0	40715.0
Medicare	33883.5	33883.5
Uninsured	37874.7	33874.7
Medicaid	33150.7	33150.7

Model m1 coefficients

```
tidy(m1, conf.int = TRUE, conf.level = 0.95) %>%
  select(term, estimate, conf.low, conf.high) %>%
  kable(digits = 2)
```

term	estimate	conf.low	conf.high
(Intercept)	33150.74	31096.68	35204.80
insuranceCommercial	7564.28	4237.15	10891.42
insuranceMedicare	732.76	-1981.74	3447.27
insuranceUninsured	4724.00	-1328.66	10776.65

- 95% CI for pop. mean `n_income` among adults with Medicaid is (31097, 35205)

Model m1 coefficients

```
tidy(m1, conf.int = TRUE, conf.level = 0.95) %>%
  select(term, estimate, conf.low, conf.high) %>%
  kable(digits = 2)
```

term	estimate	conf.low	conf.high
(Intercept)	33150.74	31096.68	35204.80
insuranceCommercial	7564.28	4237.15	10891.42
insuranceMedicare	732.76	-1981.74	3447.27
insuranceUninsured	4724.00	-1328.66	10776.65

- 95% CI for Commercial - Medicaid is (4237, 10891)
- What about Medicare - Medicaid or Uninsured - Medicaid?

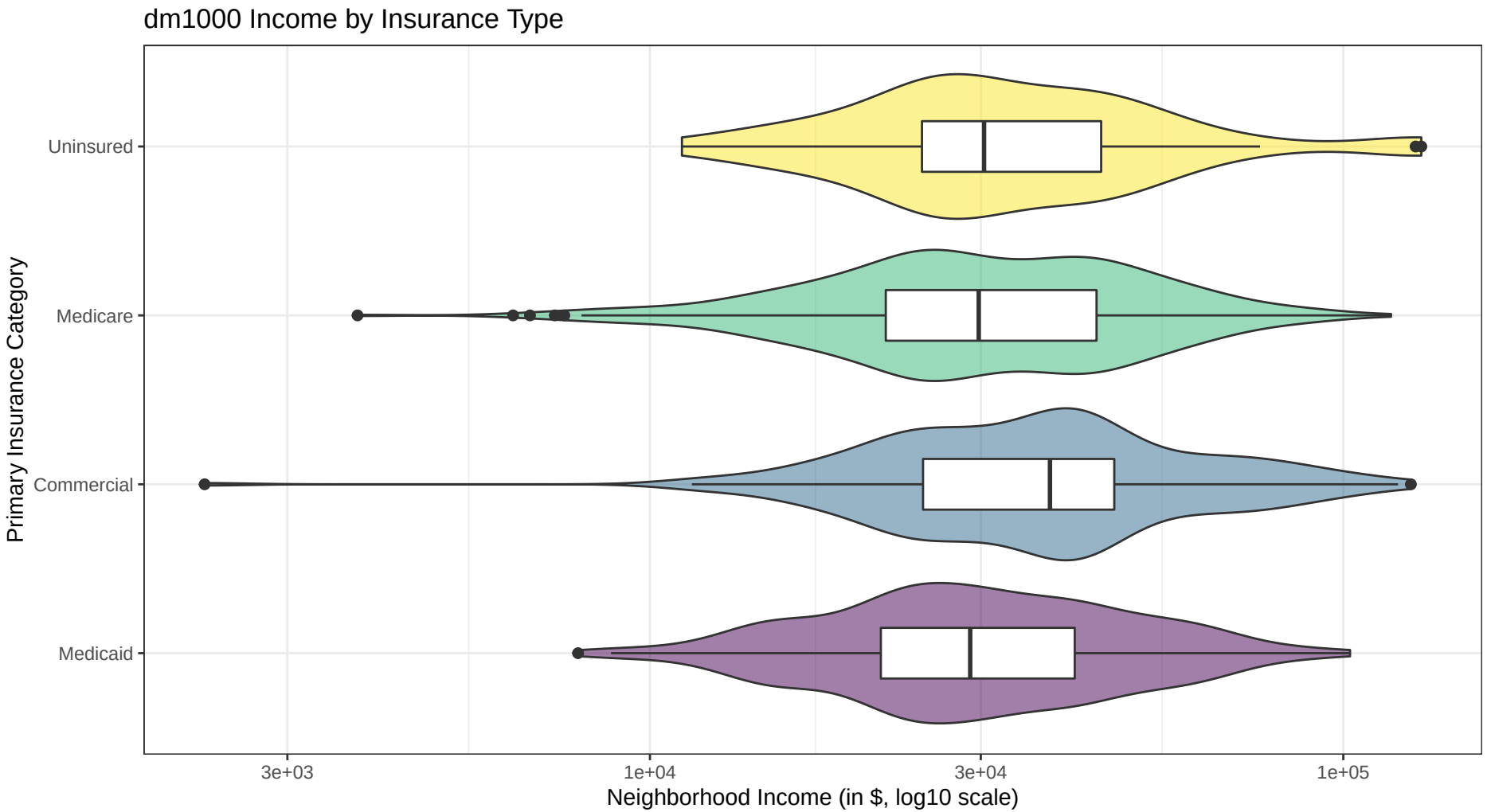
Comparing n_income across insurance groups

```
mosaic::favstats(n_income ~ insurance, data = dm1000)
```

```
   insurance   min      Q1  median       Q3    max
1   Medicaid  7876 21528.5 28965.0 40986.00 102258
2 Commercial  2279 24767.0 37748.0 46736.00 125150
3   Medicare  3787 21883.5 29797.5 44062.75 117161
4  Uninsured 11121 24678.0 30328.0 44743.00 129549
      mean       sd   n missing
1 33150.74 16814.04 315      15
2 40715.02 21713.97 194       2
3 33883.50 17529.58 422      10
4 37874.73 24962.68  41       1
```

- Does a comparison of means make sense here?
- Would it give us the same conclusions as comparing medians?

Replot on logarithmic (base 10) scale?

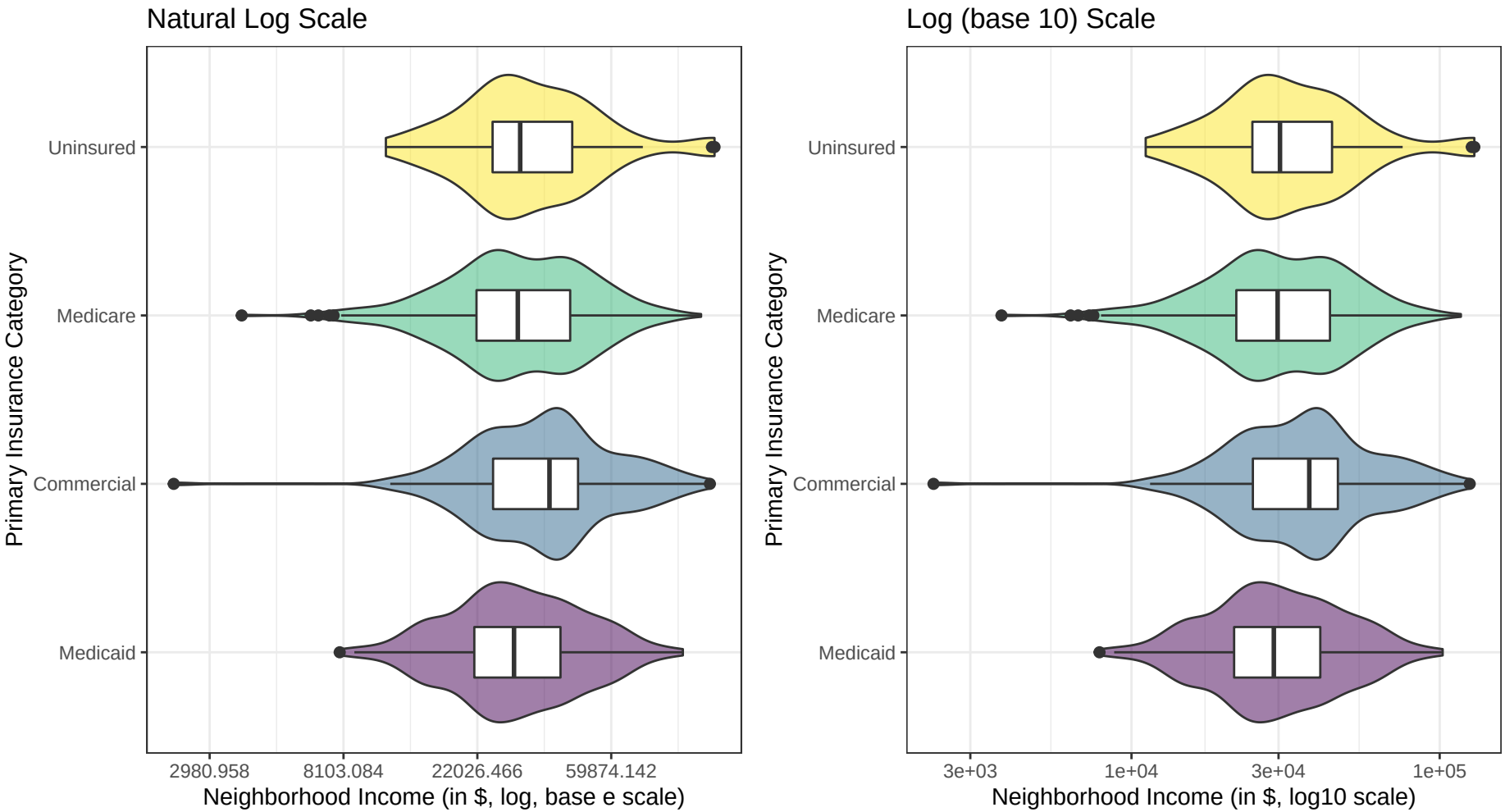


Code for plot on previous slide

```
dm1000 %>% filter(complete.cases(insurance, n_income)) %>%
  ggplot(data = ., aes(x = insurance, y = n_income)) +
  geom_violin(aes(fill = insurance)) +
  geom_boxplot(width = 0.3, outlier.size = 2) +
  guides(fill = "none") +
  coord_flip() +
  scale_fill_viridis_d(alpha = 0.5) +
  scale_y_continuous(trans = "log10") +
  labs(y = "Neighborhood Income (in $, log10 scale)",
       x = "Primary Insurance Category",
       title = "dm1000 Income by Insurance Type")
```

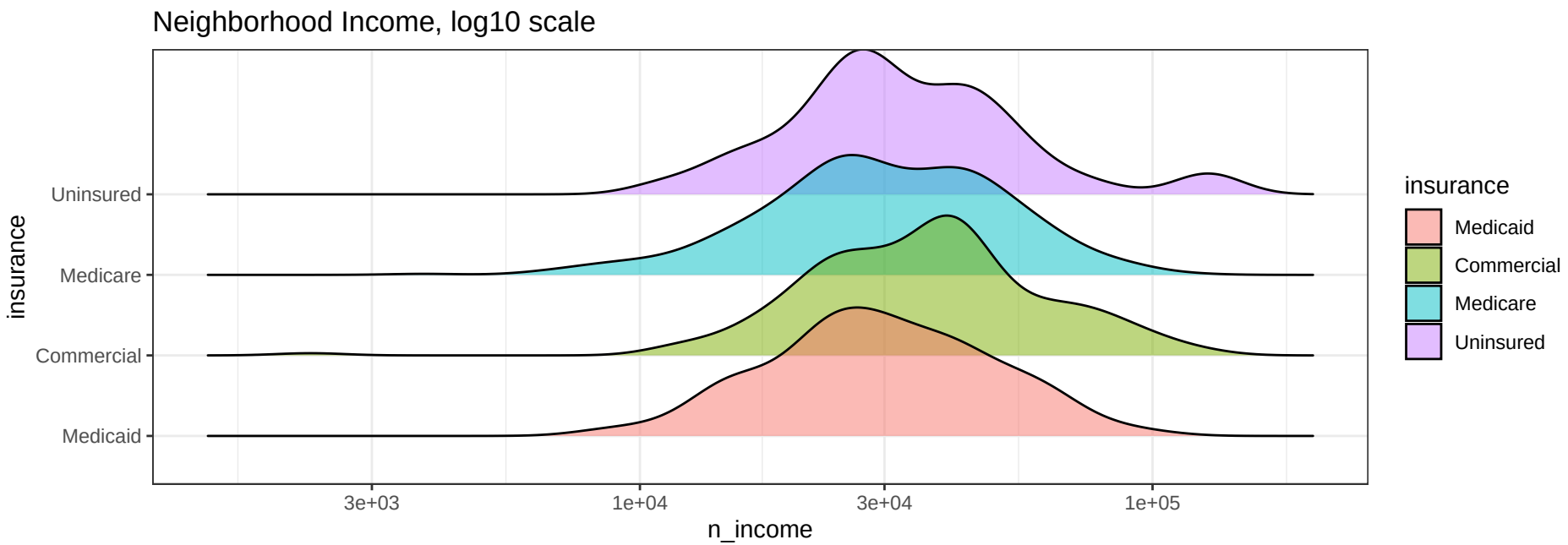
- Could also use `scale_y_log10()`

Does which logarithmic scale you pick matter?



Ridgeline Plot of Densities on Log Scale

```
dm1000 %>% filter(complete.cases(n_income, insurance)) %>%
  ggplot(data = ., aes(x = n_income, y = insurance,
                       fill = insurance)) +
  geom_density_ridges(alpha = 0.5) +
  scale_x_continuous(trans = "log10") +
  labs(title = "Neighborhood Income, log10 scale")
```



Next Up

- Favorite Movies activity
- Correlation, Association and Scatterplots