In-class 2.2, Social Science Inquiry II (SOSC13200-W23-2)

Molly Offer-Westort

Thursday 1/12/22

```
# install.packages("ggplot2") # Uncomment this the first time you run it, but
# after you have installed, comment it again.
# You only need to install a package once
library(ggplot2) # But packages you use need to be loaded every session
```

This class, we'll be exploring the Card & Krueger data and considering ways to summarize univariate (or single variable) data.

Use the ggplot2 cheat sheet for data visualizations: https://github.com/rstudio/cheatsheets/blob/main/data-visualization.pdf

Read in the Card & Krueger data.

```
file <- "https://raw.githubusercontent.com/UChicago-pol-methods/SOSC13200-W23/main/data/card-krueger.csv
dat <- read.csv(file, as.is = TRUE)
```

Look at the README file for the Card & Krueger data. This will tell you what is going on with the different variables,and how they're coded.

https://github.com/UChicago-pol-methods/SOSC13200-W23/blob/main/data/card-krueger_readme.txt

First, we'll use subsetting from last class to select the wave 1 and wave 2 data

```
# wave 1 data
dat0 <- dat[which(dat$d==0),]

# wave 2 data
dat1 <- dat[which(dat$d==1),]
```

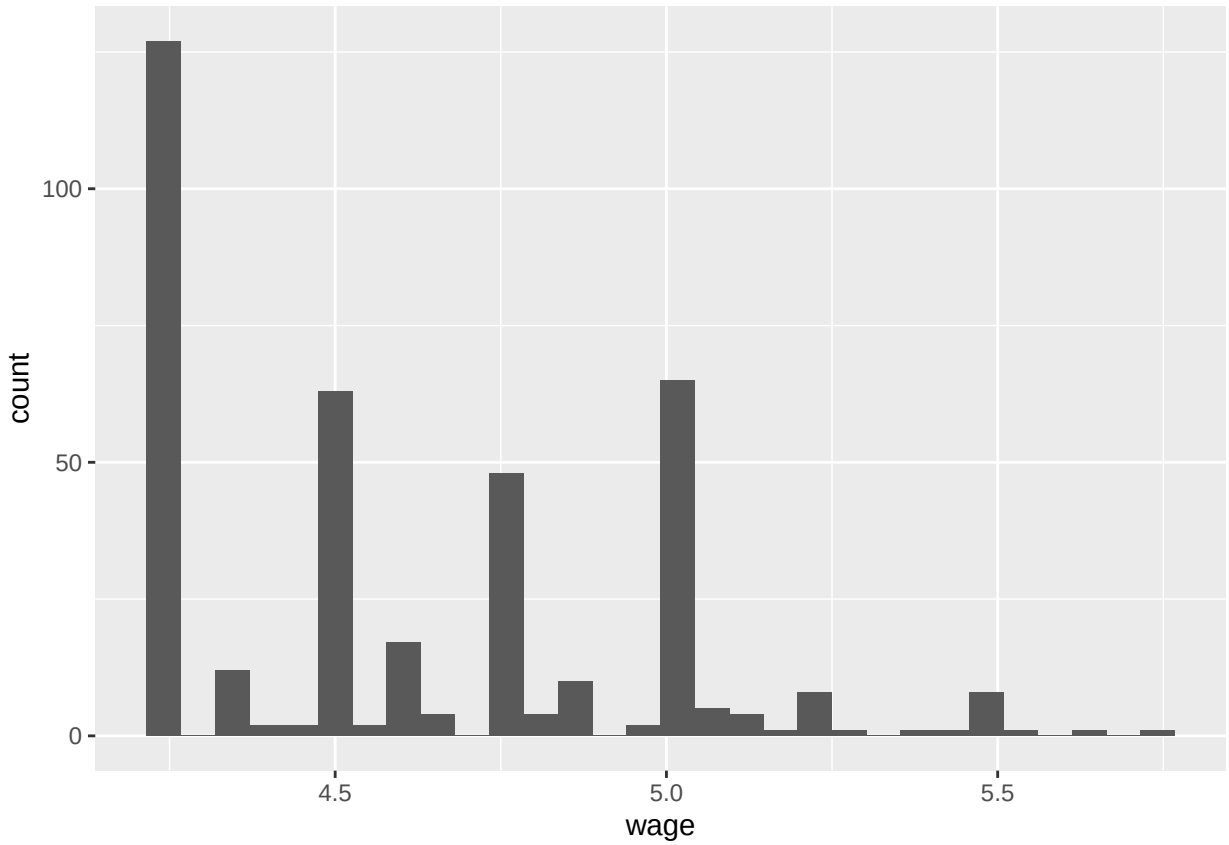
Use ggplot to look at wage date in wave 1.

There are three main components to ggplot arguments when producing data visualizations: data, instructions on aesthetic mappings, and then layers. You can add multiple layers to the same plot with the same data and mappings.

```
ggplot(dat0, # data
       aes(x = wage)) + # aesthetic mapping
  geom_histogram() # layer
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 20 rows containing non-finite values (`stat_bin()`).
```



When you run the last piece of code, we get a warning.

Should we be worried about the messages/warnings?

It depends on what the warning is telling us.

Here, it's first giving us a message "`stat_bin()` using `bins = 30`. Pick better value with `binwidth`."

This message is telling us that when we used `geom_histogram()`, ggplot set the default number of bins in the histogram plot. We can pick our own number of bins that is appropriate for our setting.

How to decide what the number of bins should be?

One answer could be that we want most unique wage values to be in their own bins.

```
unique(dat0$wage) # we can look at how many unique values of wages there are
```

```
##  [1]   NA 5.00 5.50 5.25 4.25 4.50 4.67 4.75 4.87 4.35 5.12 5.56 5.37 5.05 5.62
## [16] 5.10 5.30 5.15 5.42 5.06 5.75 4.80 4.37 4.60 4.45 4.95 4.40 4.62 4.85 4.55
## [31] 4.65 4.39 4.32
```

```
sort(unique(dat0$wage))
```

```
##  [1] 4.25 4.32 4.35 4.37 4.39 4.40 4.45 4.50 4.55 4.60 4.62 4.65 4.67 4.75 4.80
## [16] 4.85 4.87 4.95 5.00 5.05 5.06 5.10 5.12 5.15 5.25 5.30 5.37 5.42 5.50 5.56
## [31] 5.62 5.75
```

```
diff(sort(unique(dat0$wage))) # and find out how far apart unique wage values are
```

```
##  [1] 0.07 0.03 0.02 0.02 0.01 0.05 0.05 0.05 0.05 0.02 0.03 0.02 0.08 0.05 0.05
## [16] 0.02 0.08 0.05 0.05 0.01 0.04 0.02 0.03 0.10 0.05 0.07 0.05 0.08 0.06 0.06
## [31] 0.13
```

```r
# On average, the distance is about $0.05.
mean(diff(sort(unique(dat0$wage))))
```

```
## [1] 0.0483871
```

```r
# If we take the entire range of wages,
range(dat0$wage, na.rm = TRUE)
```

```
## [1] 4.25 5.75
```

```r
# and look at the distance between the lowest and highest value,
diff(range(dat0$wage, na.rm = TRUE))
```

```
## [1] 1.5
```

```r
# then divide that into length $0.05 pieces, how many pieces do we have?
diff(range(dat0$wage, na.rm = TRUE))/.05
```

```
## [1] 30
```

```r
# It's about 30-which is the default for histogram bins. So maybe 30 is fine for us.
```

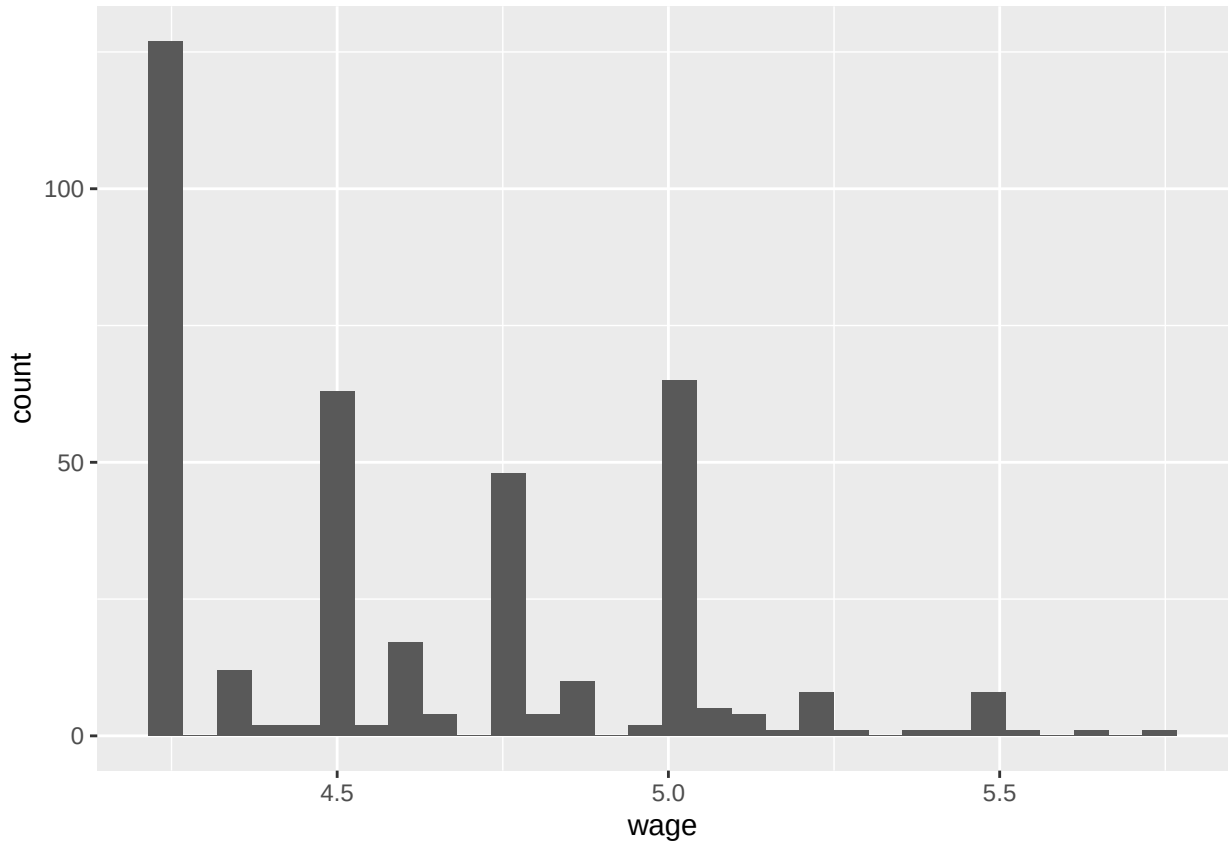
What about the warning with 20 rows of non-finite values?

That is because some of our wages are stored as missing values.

```r
table(is.na(dat0$wage))
```

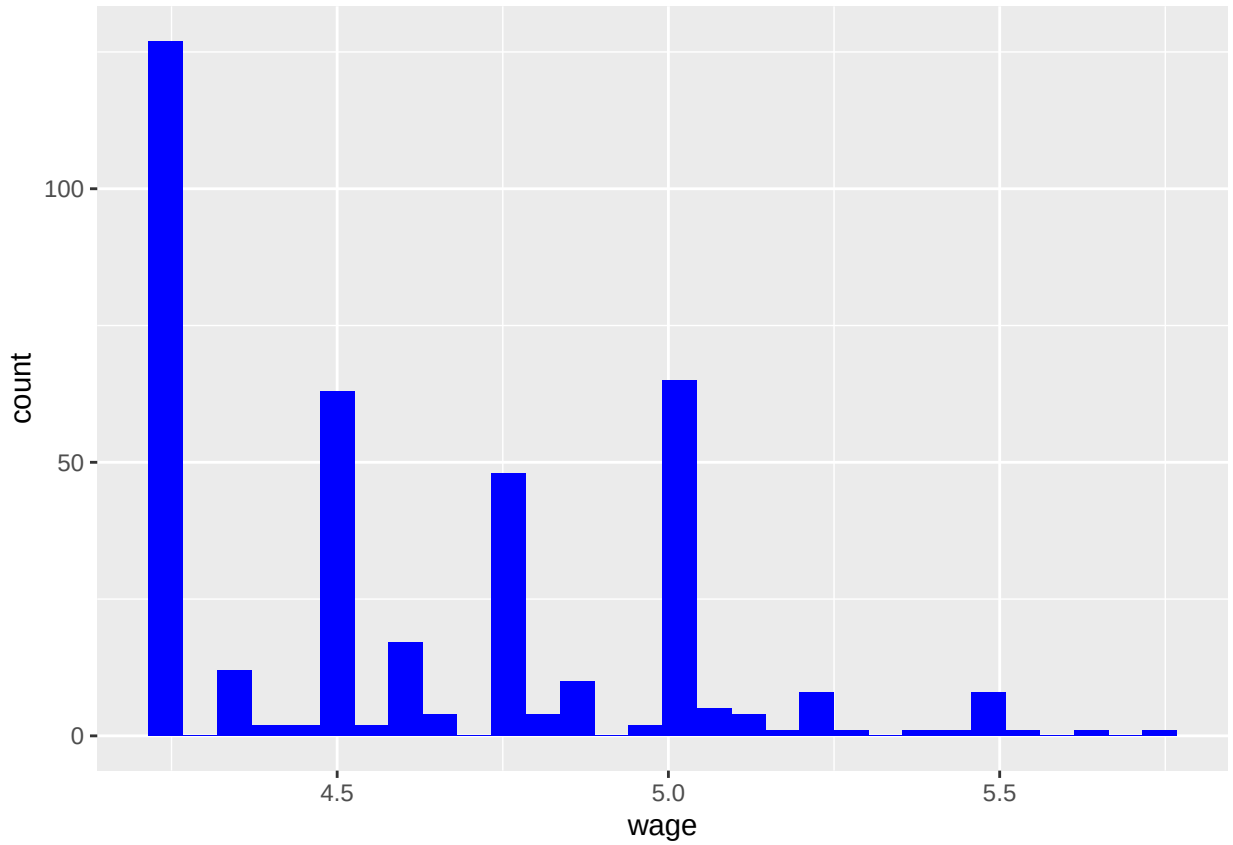
```
##
## FALSE  TRUE
##   390    20
```

```r
# We can address the warnings by setting options in the code.
ggplot(dat0, # data
       aes(x = wage)) + # aesthetic mapping
  geom_histogram(bins = 30, na.rm = TRUE) # layer
```

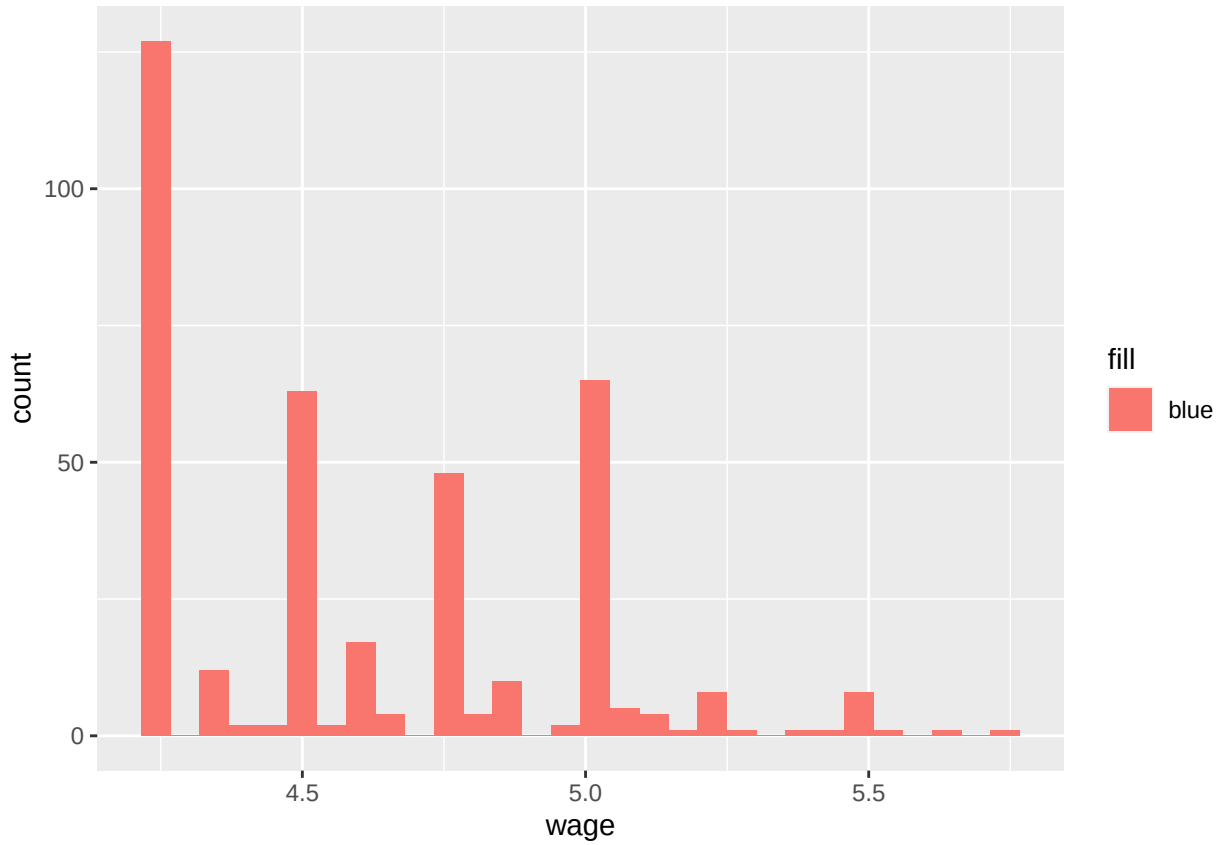


Other options

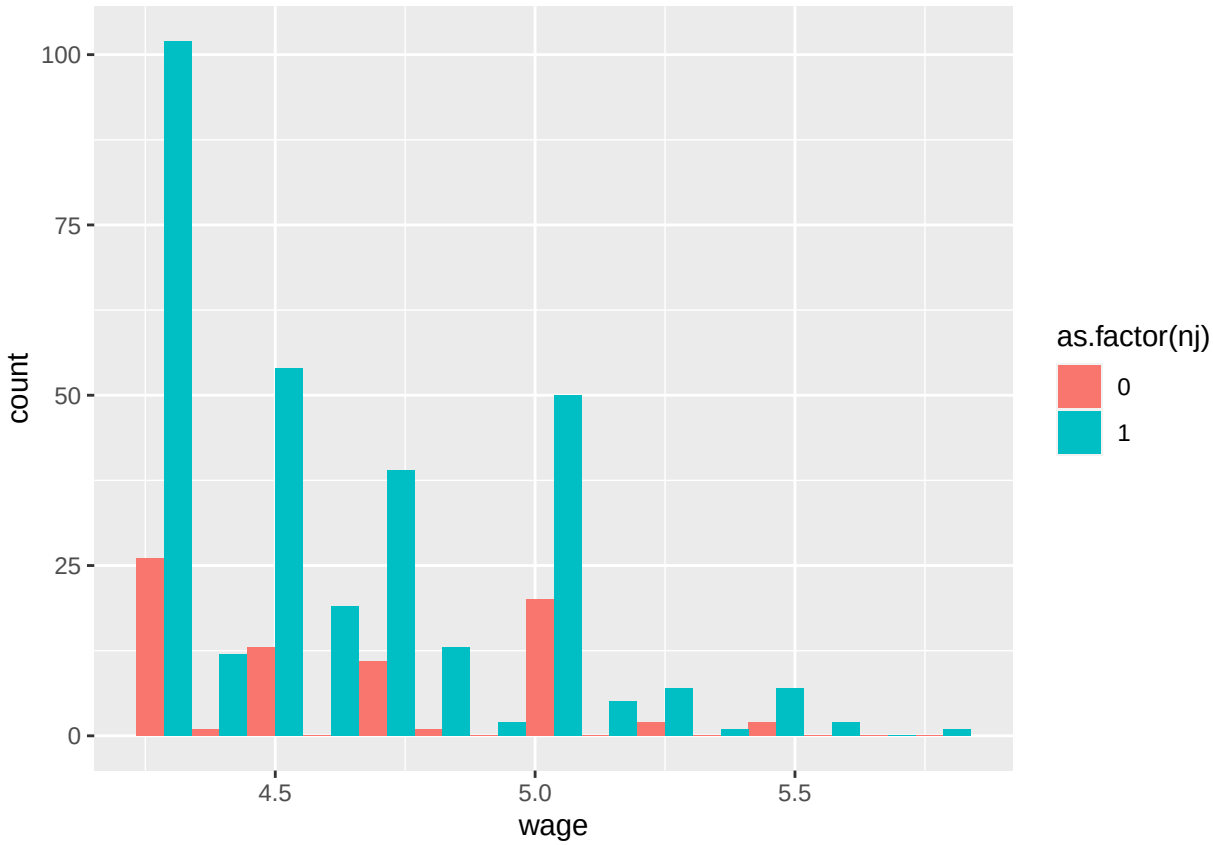
```
# we can set the color of the histogram
ggplot(dat0, # data
       aes(x = wage)) + # aesthetic mapping
  geom_histogram(bins = 30, fill = 'blue', na.rm = TRUE) # layer
```




```
# why doesn't it work here? (it's in the aesthetic, not the layer)
ggplot(dat0, # data
       aes(x = wage, fill = 'blue')) + # aesthetic mapping
  geom_histogram(bins = 30, na.rm = TRUE) # layer
```



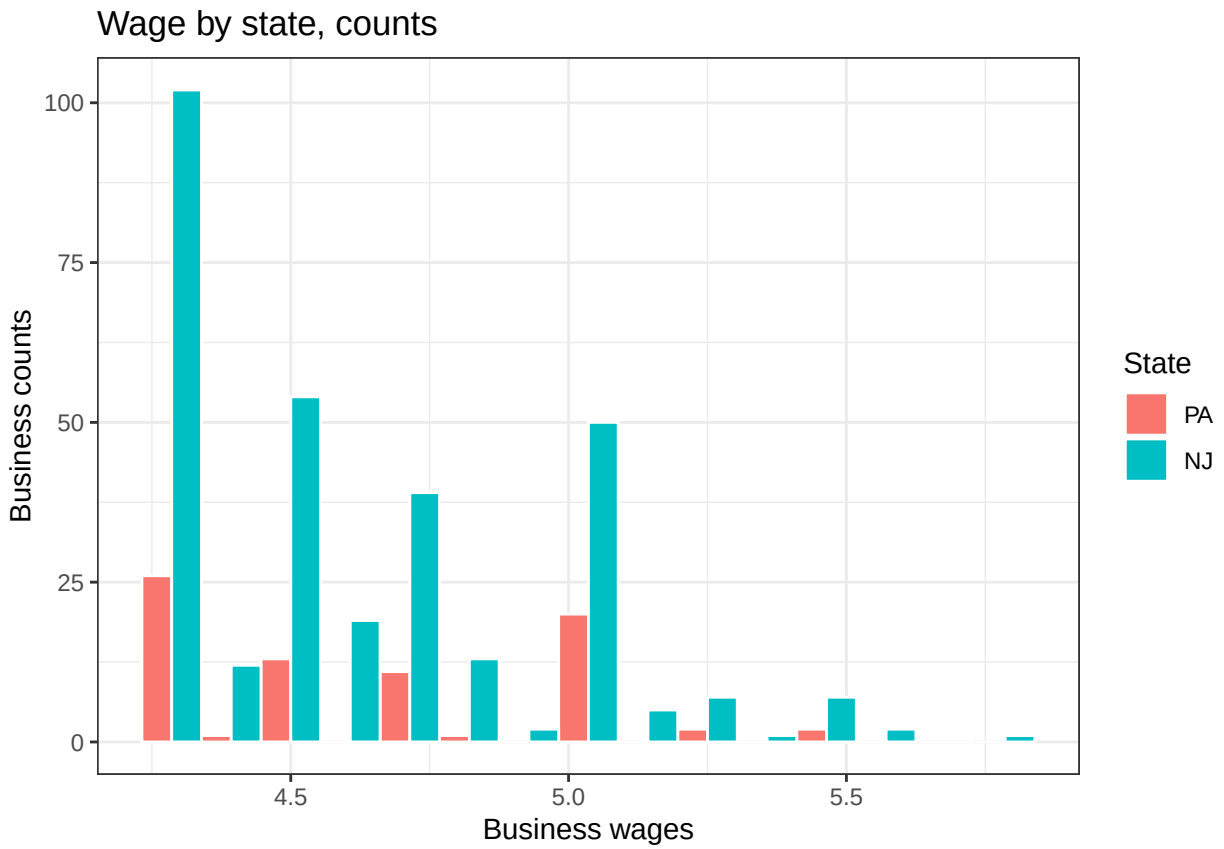

```
# we can look at pa vs. nj wages separately
ggplot(dat0, # data
       aes(x = wage, fill = as.factor(nj))) + # aesthetic mapping
  geom_histogram(bins = 15, position = 'dodge', na.rm = TRUE) # layer
```



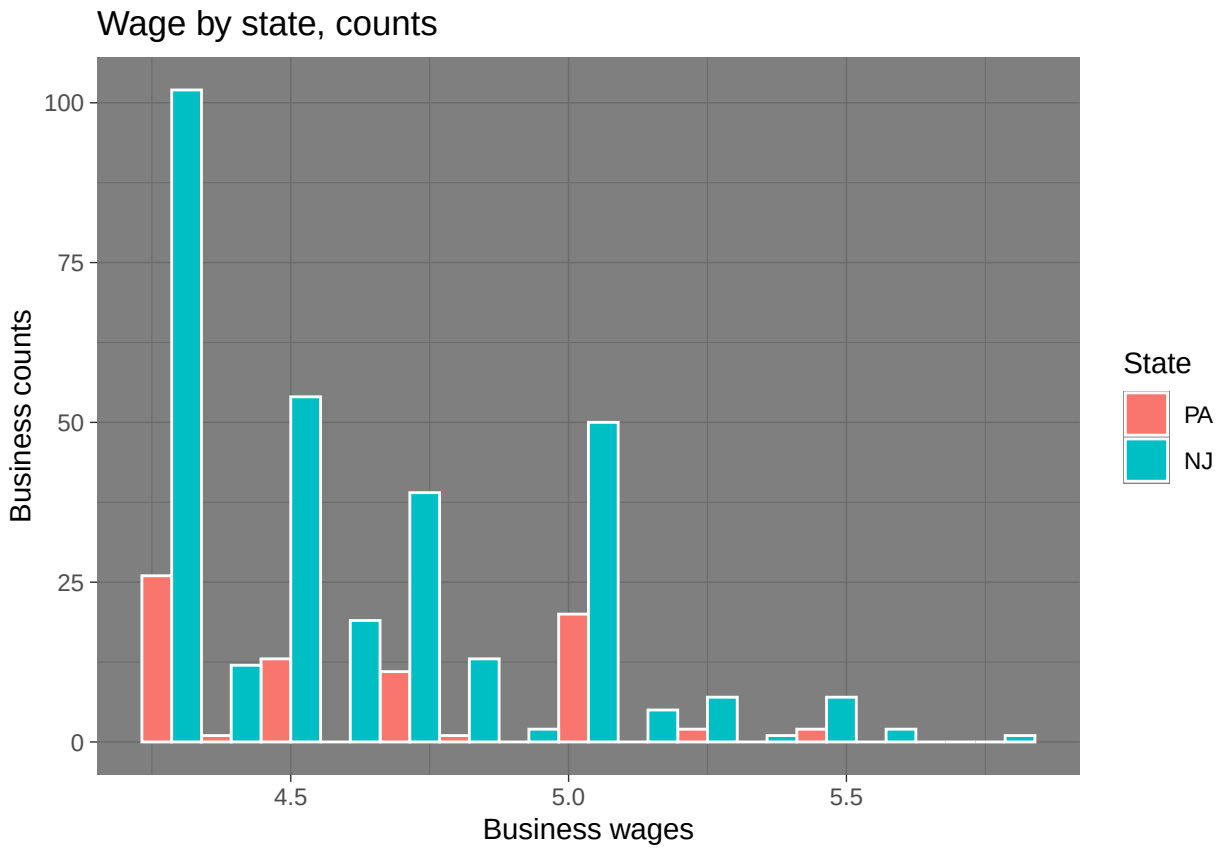

```
# labelling this in a nicer way
dat0$State <- factor(dat0$nj, labels = c('PA', 'NJ'))

g <- ggplot(dat0, # data
            aes(x = wage, fill = State)) + # aesthetic mapping
  geom_histogram(bins = 15,
                 position = 'dodge',
                 color = 'white',
                 na.rm = TRUE) + # layer
  xlab('Business wages') +
  ylab('Business counts') +
  ggtitle('Wage by state, counts')
```

Try using different themes. Note that if I save a plot as an object, I can add different layers to that object and get new plots.

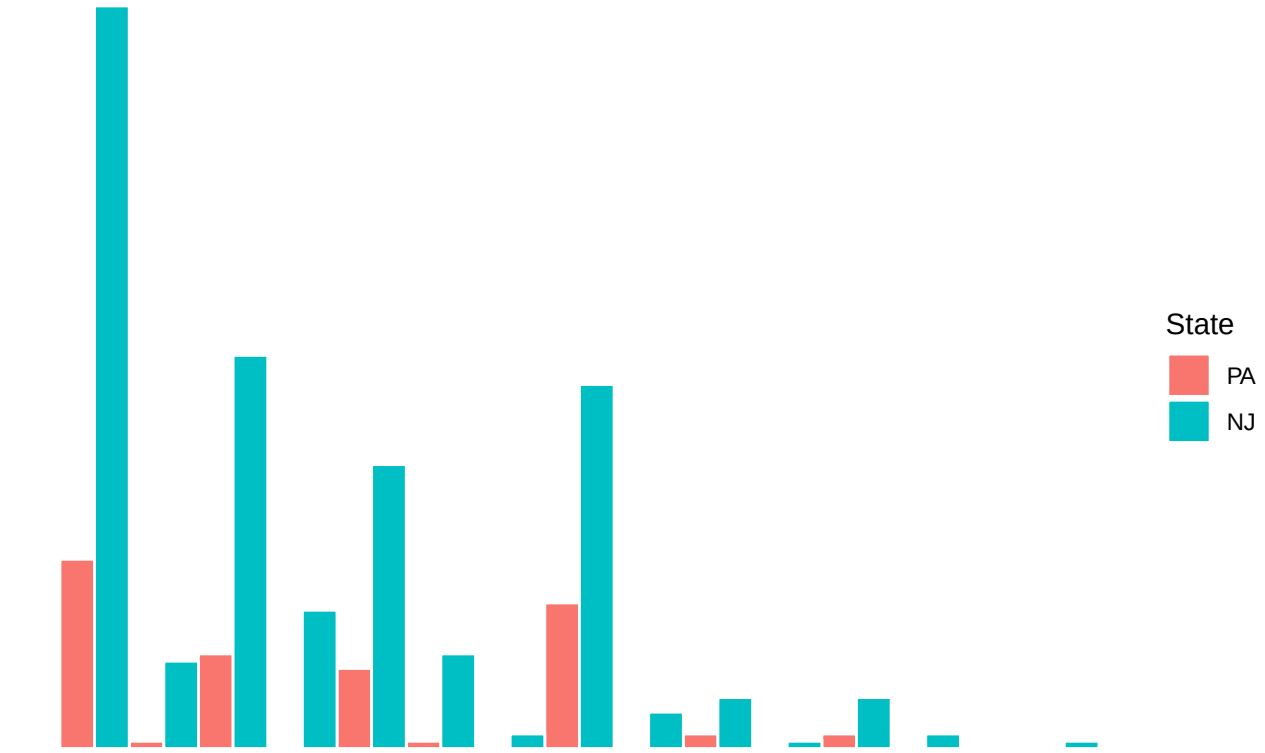
```
g + theme_bw()
```


Wage by state, counts
Business counts
100
75
50
25
0
4.5
5.0
5.5
Business wages
State
PA
NJ

```
g + theme_dark()
```


Wage by state, counts
Business counts
100
75
50
25
0
4.5
5.0
5.5
Business wages
State
PA
NJ

```
g + theme_void()
```



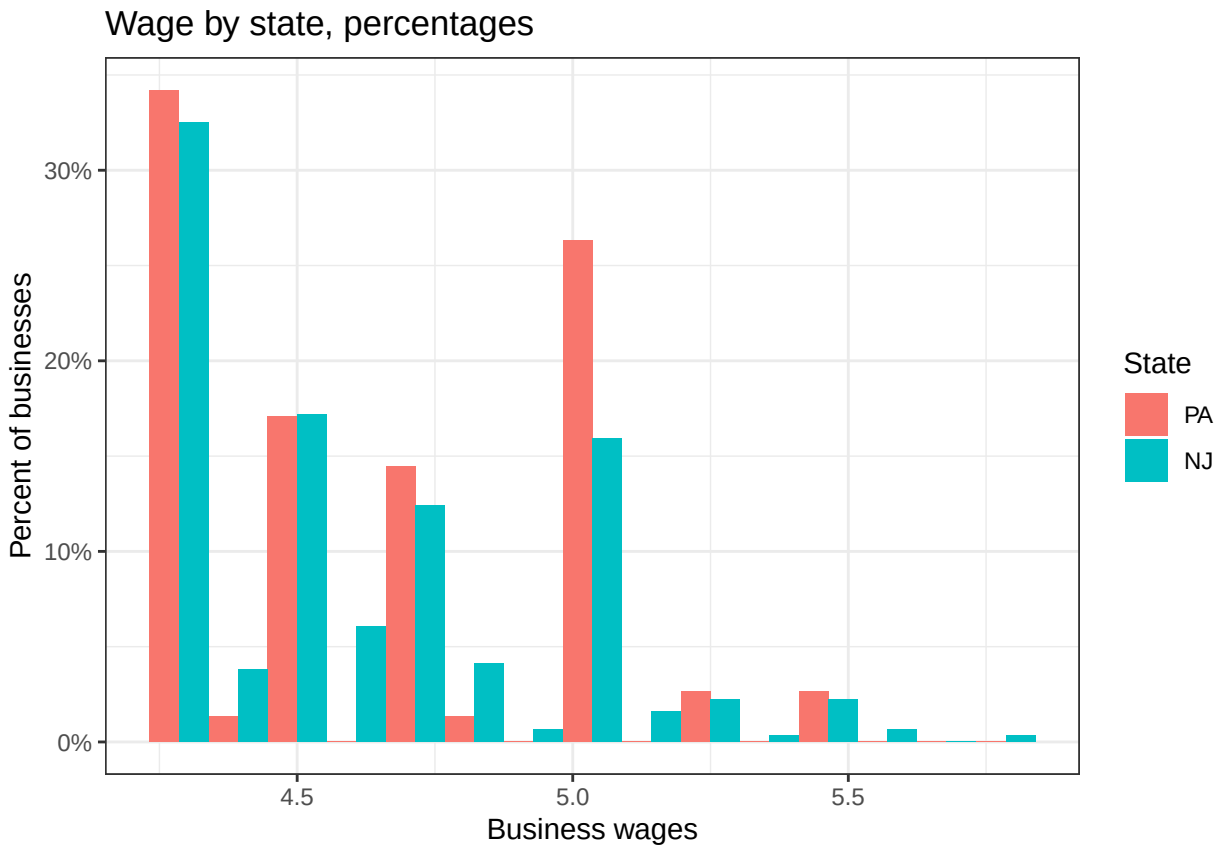

```
# you can try library(ggthemes) if you want to look up more options
```

Figure 1

I want to re-create Figure 1 from the Card and Krueger paper. To do that, I'm using the `after_stat()` function to get the y-axis to plot histograms as a percent *of each state's total*.

You don't have to worry too much about `after_stat()`, but the below illustrates how to put pieces of a plot together.

```
ggplot(dat0, aes(x = wage, fill = State)) +
  geom_histogram(aes(y=c(after_stat(count)[after_stat(group)==1]/sum(after_stat(count)[after_stat(group
                         after_stat(count)[after_stat(group)==2]/sum(after_stat(count)[after_stat(group
                 position='dodge', bins = 15, na.rm = TRUE) +
  scale_y_continuous(labels = scales::percent) +
  theme_bw() +
  xlab('Business wages') +
  ylab('Percent of businesses') +
  ggtitle('Wage by state, percentages')
```



Using facet wrap to show the two states side-by-side

```
g + facet_wrap(~State)
```


```
# letting the two plots have different y-axes
g + facet_wrap(vars(State), scales = 'free_y')
```

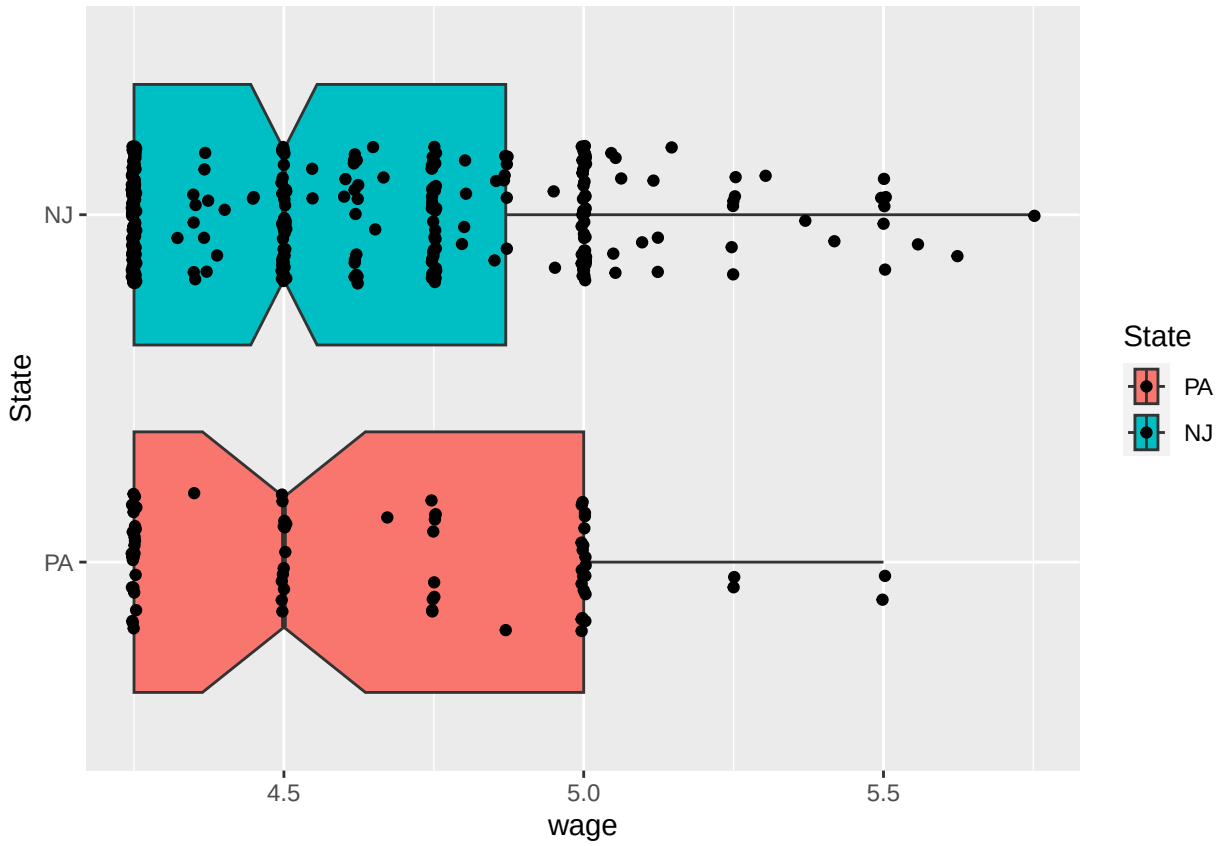

```
# changing how the x-axis is plotted
g + facet_wrap(vars(State), scales = 'free_y')+
  theme(axis.text.x = element_text(angle = 45, vjust = -1, hjust = 0.25))
```

Boxplots Boxplots are a very useful way to visualize univariate data. Here, I have overlayed it with the actual data (with jitters, so I can see every data point).

A nice data visualization practice is, in addition to any summary plots you present, see if you can also present all of the individual data points.

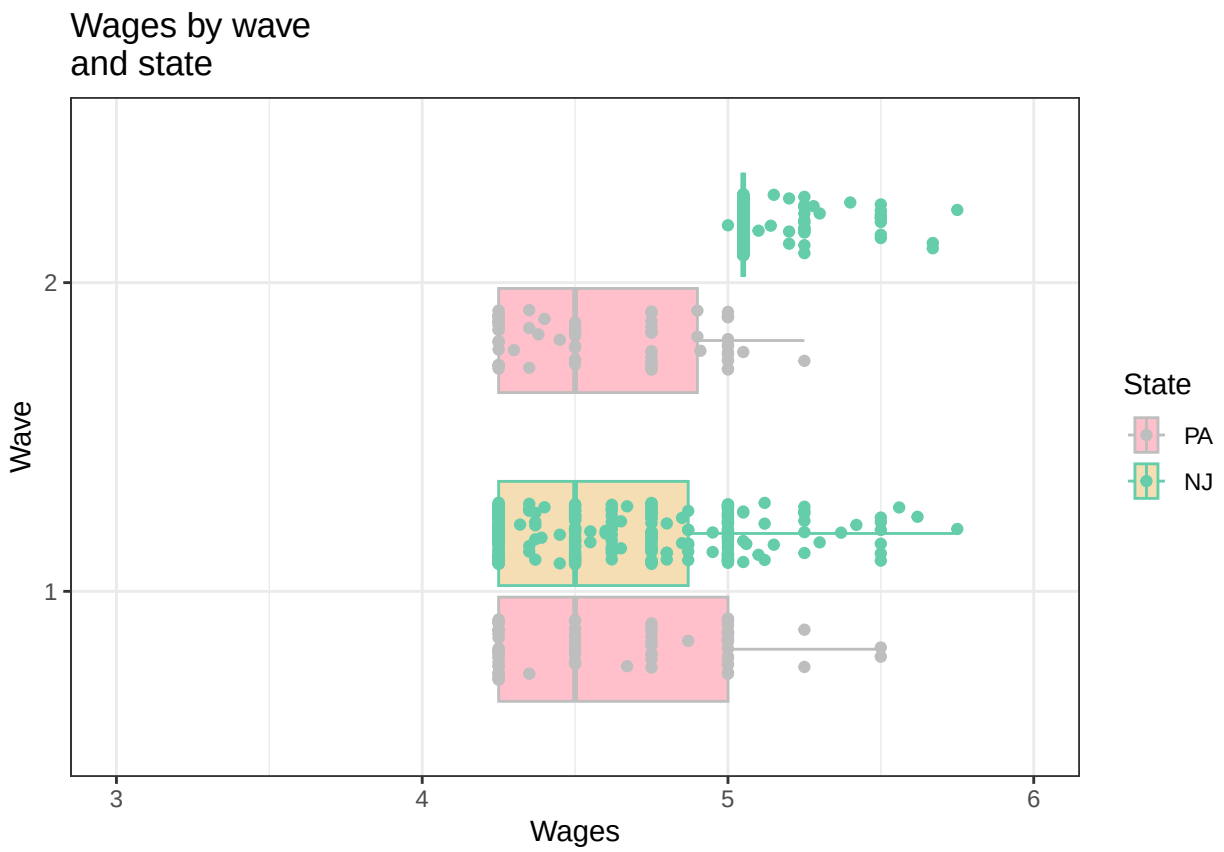
Sometimes there are too many data points to do this nicely.

```
ggplot(dat0, # data
       aes(x = wage, y = State, fill = State)) +
  geom_boxplot(notch = TRUE, na.rm = TRUE) +
  geom_jitter(height = 0.2, na.rm = TRUE)
```



What do we see about the range of NJ wages vs. the range of PA wages?

```
# Looking at this by state (with some extra visualizations thrown in)
dat$State <- factor(dat$nj, labels = c('PA', 'NJ'))

ggplot(dat, # data
       aes(x = wage, y = as.factor(d+1), fill = State, color = State)) +
  scale_fill_manual(values=c("pink", "wheat")) +
  scale_color_manual(values=c("grey", "mediumaquamarine")) +
  geom_boxplot(outlier.shape = NA, na.rm = TRUE)+
  geom_point(position=position_jitterdodge(), na.rm = TRUE) +
  theme_bw() +
  coord_cartesian(xlim = c(3, 6)) +
  xlab('Wages') +
  ylab('Wave') +
  ggtitle('Wages by wave\nand state')
```



We can also compare the distributions numerically, using the `summarize()` function.

```
# NJ data in wave 1
summary(dat$wage[which(dat$d == 0 & dat$State == 'NJ')])
```

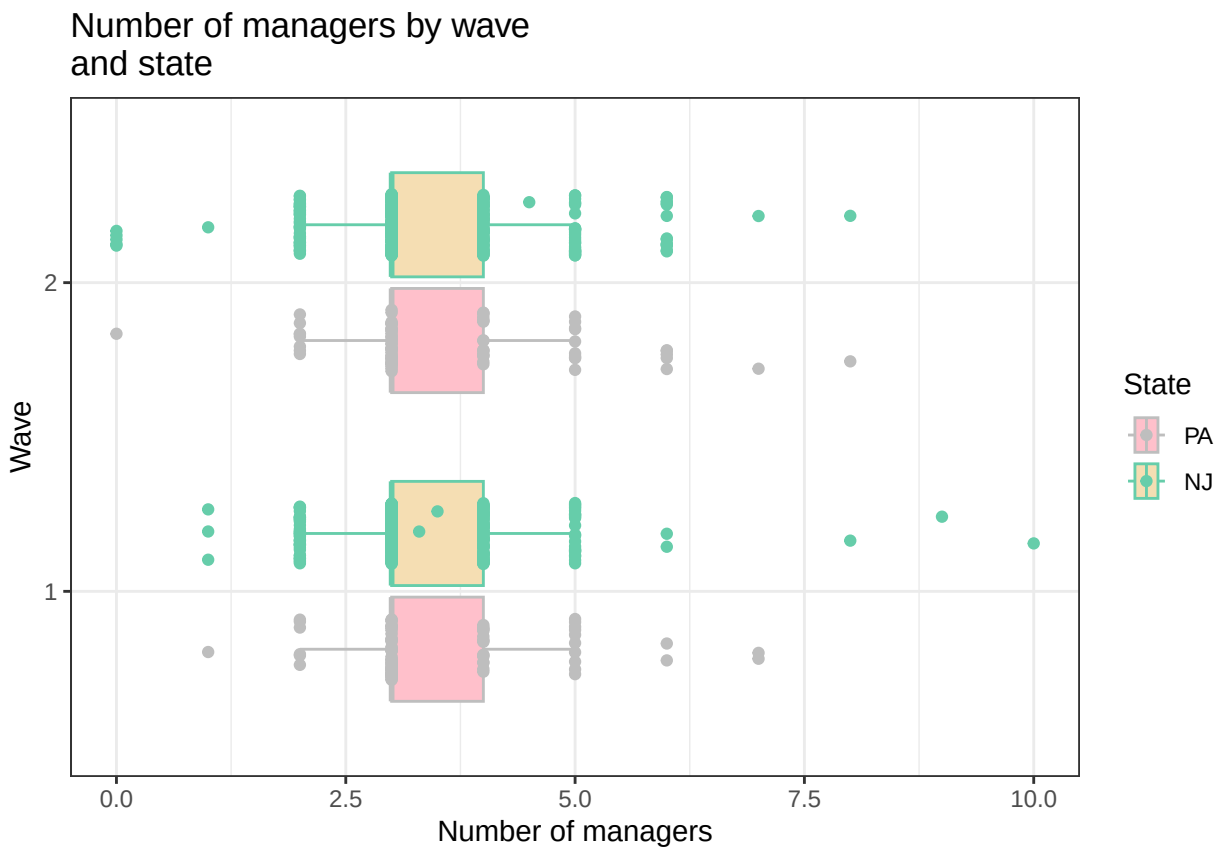
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   4.250   4.250   4.500   4.612   4.870   5.750      17
```

```
# NJ data in wave 2
summary(dat$wage[which(dat$d == 1 & dat$State == 'NJ')])
```

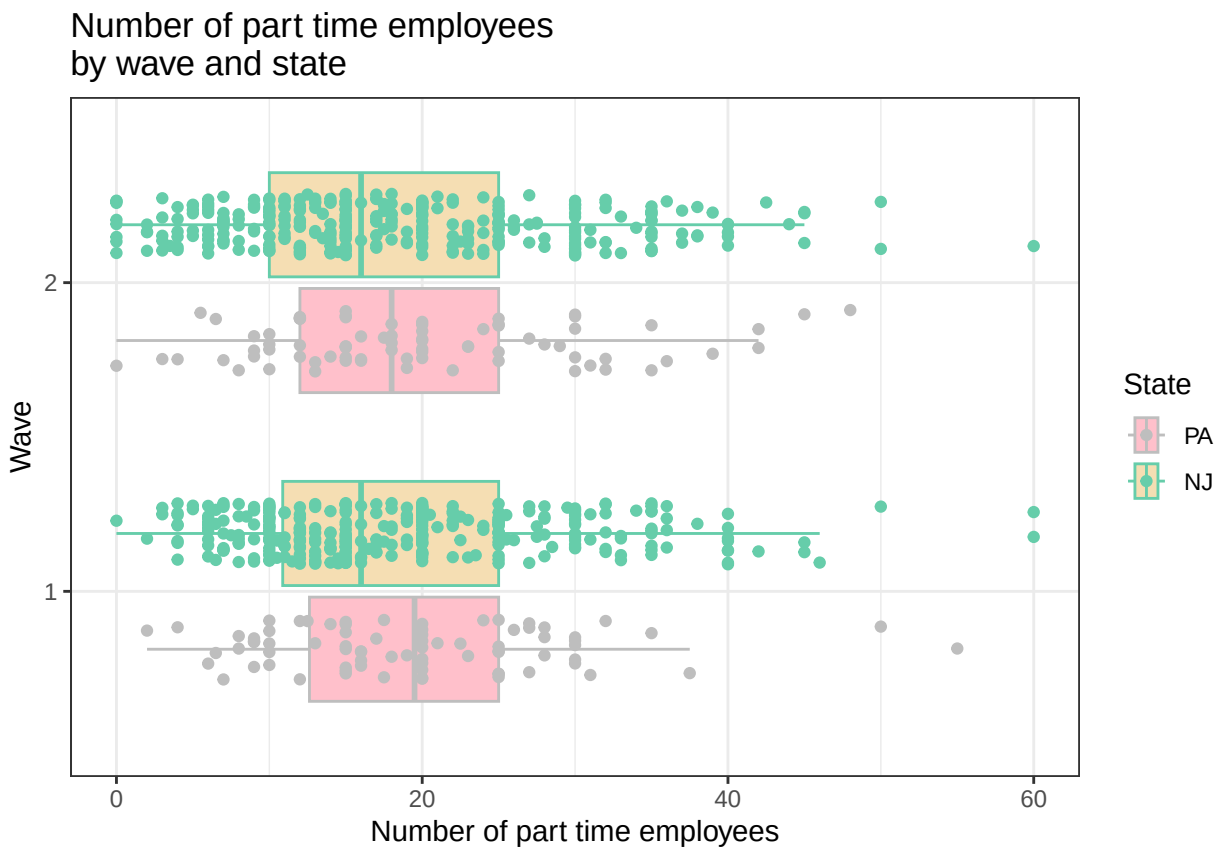
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   5.000   5.050   5.050   5.081   5.050   5.750      13
```

Exploring how staffing changes by state and wave

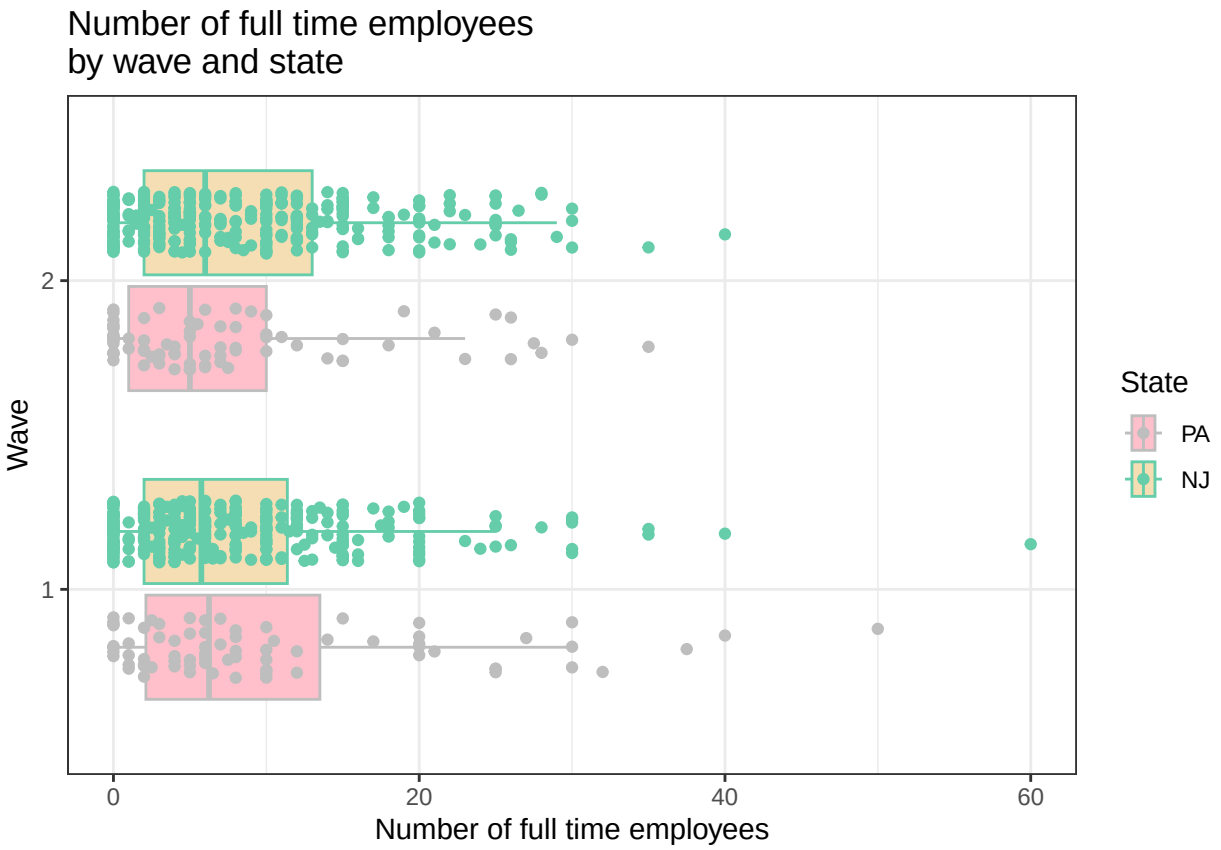
```
ggplot(dat, # data
       aes(x = mgrs, y = as.factor(d+1), fill = State, color = State)) +
  scale_fill_manual(values=c("pink", "wheat")) +
  scale_color_manual(values=c("grey", "mediumaquamarine")) +
  geom_boxplot(outlier.shape = NA, na.rm = TRUE)+
  geom_point(position=position_jitterdodge(), na.rm = TRUE) +
  theme_bw() +
  xlab('Number of managers') +
  ylab('Wave') +
  ggtitle('Number of managers by wave\nand state')
```




```
ggplot(dat, # data
       aes(x = pt, y = as.factor(d+1), fill = State, color = State)) +
  scale_fill_manual(values=c("pink", "wheat")) +
  scale_color_manual(values=c("grey", "mediumaquamarine")) +
  geom_boxplot(outlier.shape = NA, na.rm = TRUE)+
  geom_point(position=position_jitterdodge(), na.rm = TRUE) +
  theme_bw() +
  xlab('Number of part time employees') +
  ylab('Wave') +
  ggtitle('Number of part time employees\nby wave and state')
```




```
ggplot(dat, # data
       aes(x = ft, y = as.factor(d+1), fill = State, color = State)) +
  scale_fill_manual(values=c("pink", "wheat")) +
  scale_color_manual(values=c("grey", "mediumaquamarine")) +
  geom_boxplot(outlier.shape = NA, na.rm = TRUE)+
  geom_point(position=position_jitterdodge(), na.rm = TRUE) +
  theme_bw() +
  xlab('Number of full time employees') +
  ylab('Wave') +
  ggtitle('Number of full time employees\nby wave and state')
```



Exercises

Run `vignette("ggplot2-specs")` in your console to get an overview of some ggplot2 aesthetics that can be modified.

Make a boxplot that shows number of full time equivalent employees by wave and state. Look up `?geom_boxplot` and modify several different options. Try seeing what shapes you can use for outliers.

Make a histogram that shows number of hours open per day, by co-owned status. Use colors to show an additional aesthetic component. Try faceting with two different variables.

Recreate one of the boxplots from above, using a violin plot (`geom_violin`). Does this give you different information? See what it looks like overlaying geom_point() or geom_jitter().