Lecture 4 Exercises

Isabel Fulcher

8/9/2018

Load packages

```
library("tidyverse") #ggplot2, dplyr, etc.
library("reshape2") #need this for melt()
library("car") #contains dataset
```

Part I: Basic data analysis using tidyverse

Open the Blackmore dataset in R. This contains data on 138 teenaged girls hospitalized for eating disorders and 98 control subjects. There are four variables: subject id, age in years, hours per week of exercise, and group (control vs. eating-disorder). There are multiple observations per subject.

```
data(Blackmore)
head(Blackmore)
```

```
##   subject   age exercise   group
## 1     100  8.00     2.71 patient
## 2     100 10.00     1.94 patient
## 3     100 12.00     2.36 patient
## 4     100 14.00     1.54 patient
## 5     100 15.92     8.63 patient
## 6     101  8.00     0.14 patient
```

Key commands

Pick observations by their values with filter()

```
Blackmore %>% filter(group=="patient")

# You do not have to use piping, just makes it cleaner
filter(Blackmore,group=="patient")

# You can also just use base R
Blackmore[Blackmore$group=="patient",]
```

Pick variables by their names with select()

```
Blackmore %>% select(subject,exercise)

#and in base R...
Blackmore[c("subject","exercise")]
```

Create new variables with functions of existing variables with mutate()

```
Blackmore %>% mutate(age_new = age^2)

#and in base R...
Blackmore$age_new = Blackmore$age^2
```

Collapse many values down to a single summary with summarise()

```
Blackmore %>% summarise(age_mean = mean(age),age_sd = sd(age))

#and in base R...
mean(Blackmore$age)
sd(Blackmore$age)
```

To only perform the summarise command on a subset use group_by()

```
Blackmore %>% group_by(group) %>% summarise(age_mean = mean(age),age_sd = sd(age))

#and in base R...
mean(Blackmore[Blackmore$group=="patient",]$age)
mean(Blackmore[Blackmore$group=="patient",]$age)
sd(Blackmore[Blackmore$group=="control",]$age)
sd(Blackmore[Blackmore$group=="patient",]$age)
```

Plotting with ggplot2

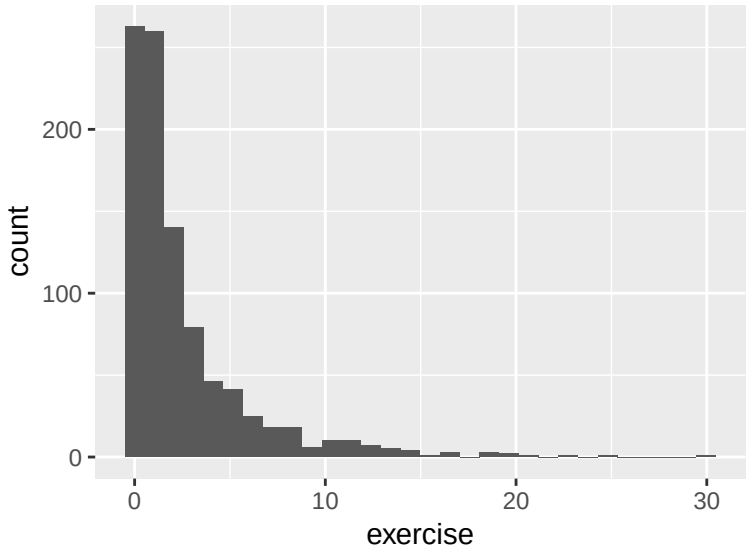
The basic idea

```
# Initialize the plot with data and aesthetics
p <- ggplot(data=NULL,mapping=aes(x=,y=,fill=,group=,color=))

# Add layers
p <- p + geom_histogram()

# Make pretty
p <- p + theme_bw() + labs(title="",x="",y="")
```

Example: Plot a histogram of hours per week of exercise across all observations.

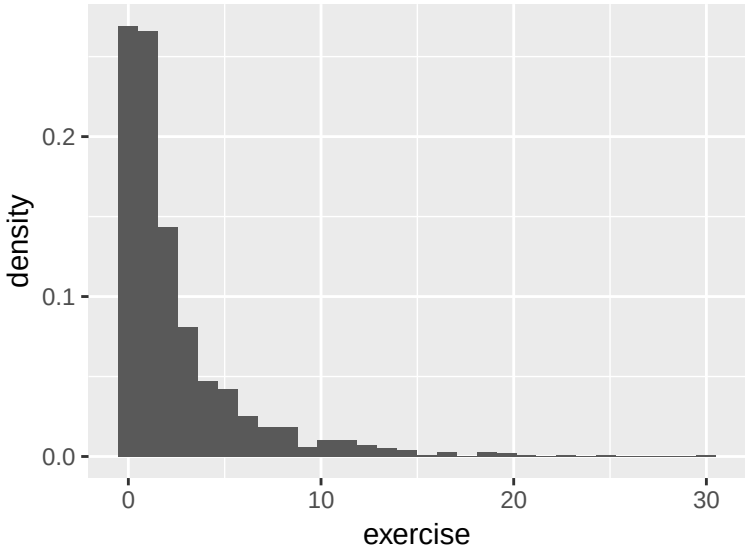
```
# Initialize the plot with data and aesthetics
p <- ggplot(Blackmore,aes(x=exercise))

# Add layer
p1 <- p + geom_histogram()
p1
```

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



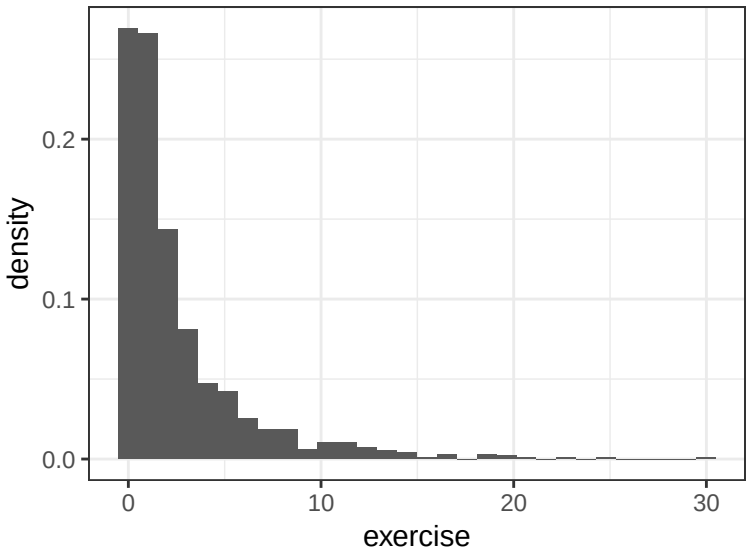

```
p2 <- p + geom_histogram(aes(y=stat(density)))
p2
```

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




```
# Make pretty
p2 + theme_bw()
```

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



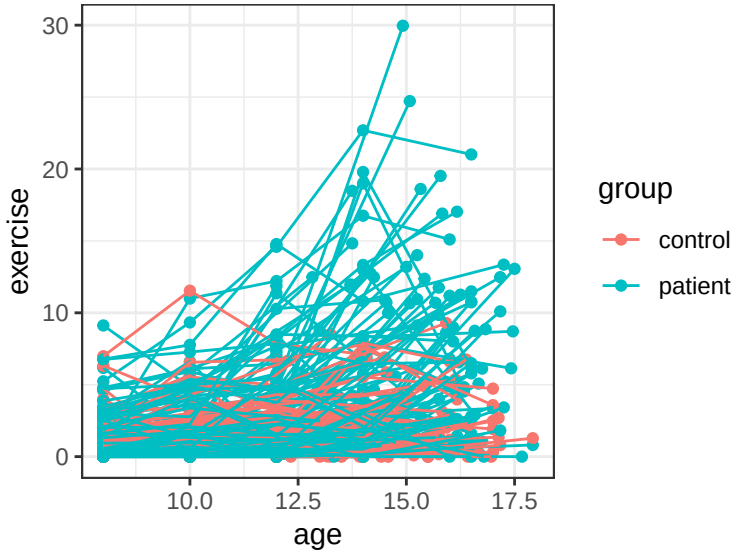
Saving graphics

```
# Setting your working directory
setwd("your/file/path/here")

# Use ggsave
ggsave("plot_name.pdf",plot = p)
```

Part I: Exercises

1. Plot a line for each person that shows exercise at each age observation. Color the line by group.

```
#correct
ggplot(Blackmore,aes(x=age,y=exercise,group=subject,color=group)) +
  geom_point() + geom_line() + theme_bw()
```



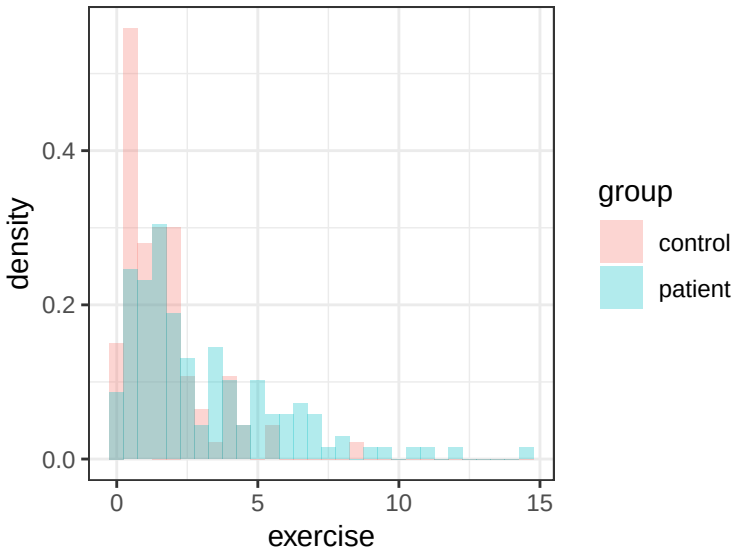
2. Create a new data frame that only contains the subject id, group, and average exercise per person.

```
Blackmore %>% dplyr::select(subject,group,exercise) %>%
  group_by(subject,group) %>% summarise(exercise=mean(exercise)) -> df
```

3. Using the data frame from question 2, plot two histograms of mean hours per week of exercise by group in the same figure.

```
ggplot(df,aes(exercise,fill=group)) +
  geom_histogram(aes(y=stat(density)),alpha=0.3,position='identity') + theme_bw()
```

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



4. Calculate the 95% confidence interval for the mean exercise hours for each group. The confidence interval is given by,

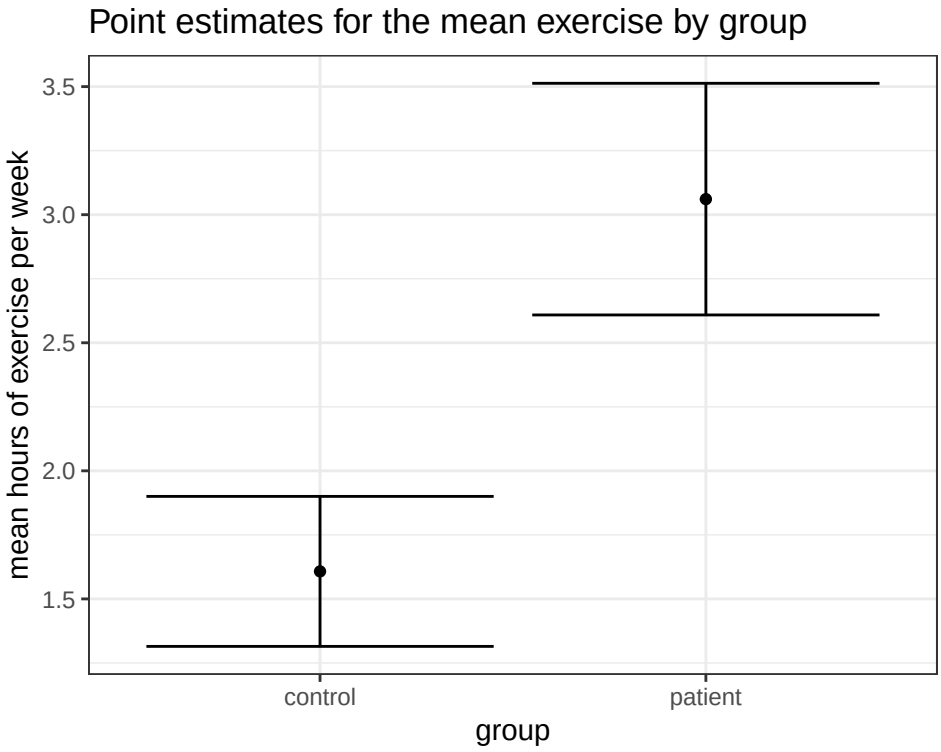
$$(\bar{X} - z_{.975}\sigma/\sqrt{n}, \bar{X} + z_{.975}\sigma/\sqrt{n})$$

To calculate $z_{.975}$, use a command we learned last class (hint: starts with a 'q' and ends with a 'norm'). You can use the estimated standard deviation in place of σ (for those of you who recognize that this is naughty, I'm unconcerned in this case becase n is large in each group).

```
Blackmore %>% dplyr::select(subject,group,exercise) %>%
  group_by(subject,group) %>%
  summarise(meanex1=mean(exercise)) %>%
  group_by(group) %>% summarise(meanex=mean(meanex1),sd=sd(meanex1),n=n()) %>%
  mutate(ci_low = meanex-qnorm(.975)*sd/sqrt(n),
         ci_up = meanex+qnorm(.975)*sd/sqrt(n)) -> df
```

5. Plot these two confidence intervals on the same plot using geom_errorbar()

```
ggplot(df,aes(y=meanex,x=group)) +
  geom_point() + geom_errorbar(aes(ymin=ci_low, ymax=ci_up)) +
  theme_bw() + labs(title="Point estimates for the mean exercise by group",y="mean hours of exercise pe
```


Point estimates for the mean exercise by group
mean hours of exercise per week
3.5
3.0
2.5
2.0
1.5
control
patient
group

Part II: Central Limit Theorem (CLT) exercise

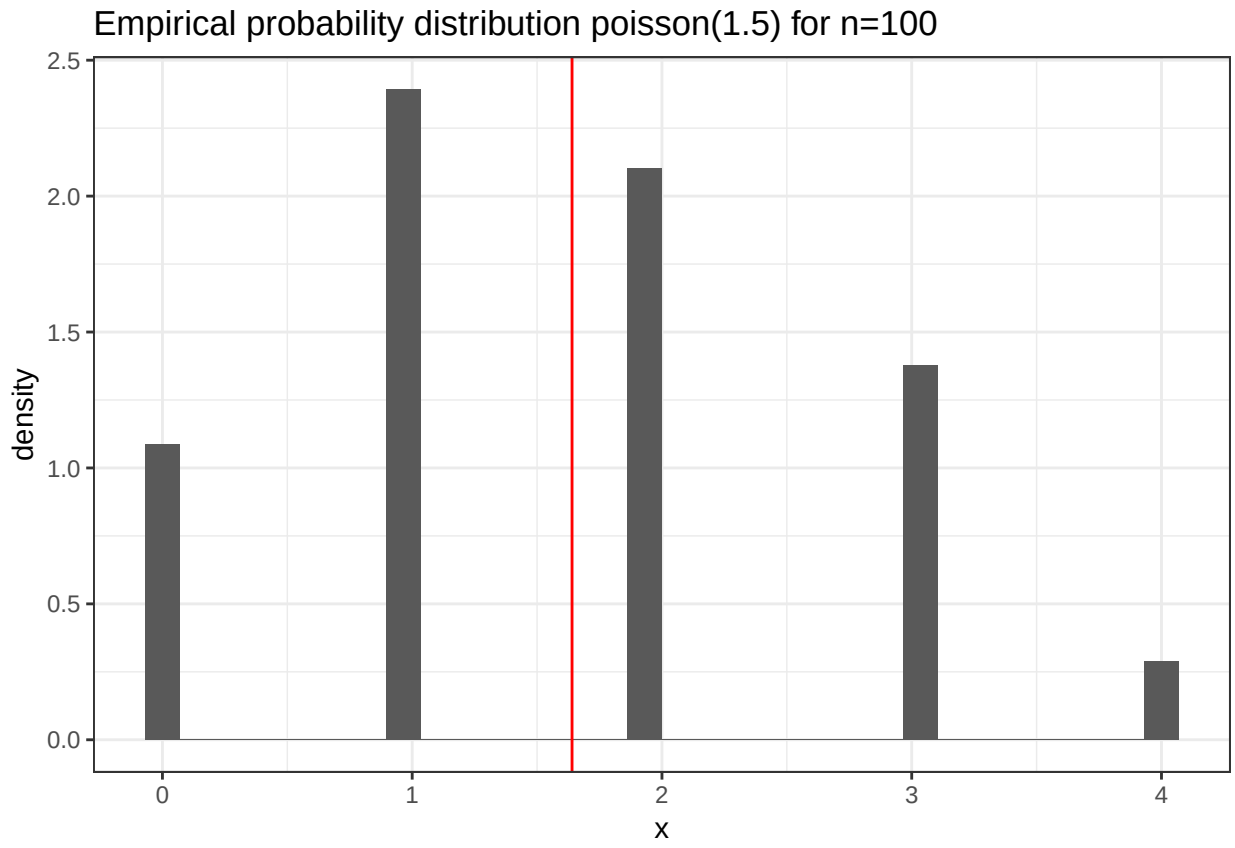
The goal here is to illustrate the Central Limit Theorem. The Central Limit Theorem states that the sample mean will converge in distribution to a normal distribution. Given $X_1, ...X_n$ independent draws from some underlying distribution,

$$\frac{(\bar{X}_n - \mu)}{\sigma/\sqrt{n}} \rightarrow N(0,1)$$

1. Draw one random sample of size 100 from a Poisson distribution with event rate λ equal to 1.5. Use a histogram to display the empirical probability distribution. Include a vertical line indicating the sample mean using geom_vline().

```
set.seed(1567)
lambda <-  1.5
x <- rpois(100,1.5)
x.df <- data.frame(x)

ggplot(x.df,aes(x)) + geom_histogram(aes(y= stat(density))) +
  geom_vline(xintercept=mean(x),col="red") + #add red vertical line
  theme_bw() +
  labs(title="Empirical probability distribution poisson(1.5) for n=100") +
  scale_x_continuous(breaks=seq(0,max(x),1)) #change the number of ticks
```

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



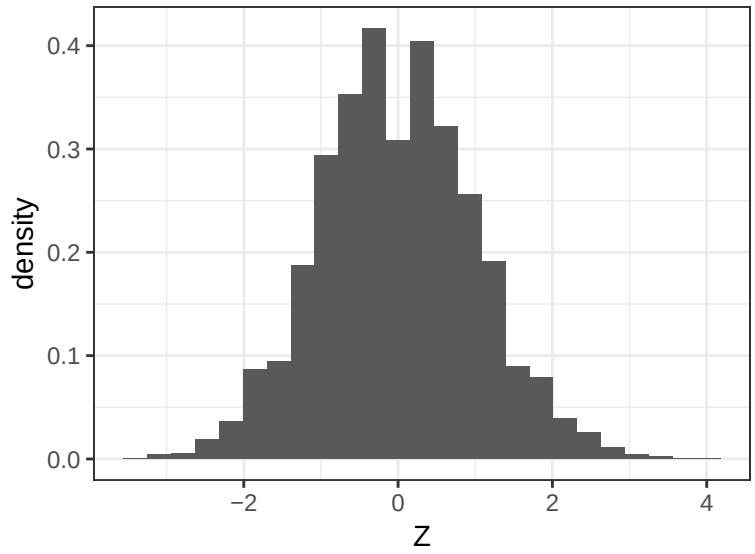
2. Draw 10,000 random samples of size 100 from a Poisson distribution with event rate λ equal to 1.5.

Calculate the quantity on the left hand side of the above equation for each sample. Note that the poisson distribution has mean $E[X] = \mu = \lambda$ and $Var[X] = \sigma^2 = \lambda$.

```
set.seed(33)
R = 10000
sigma = sqrt(1.5)
Z <- sapply(1:R,function(x,n) {sqrt(n)*(mean(rpois(n,lambda)) - lambda)/sigma},n=100)
```

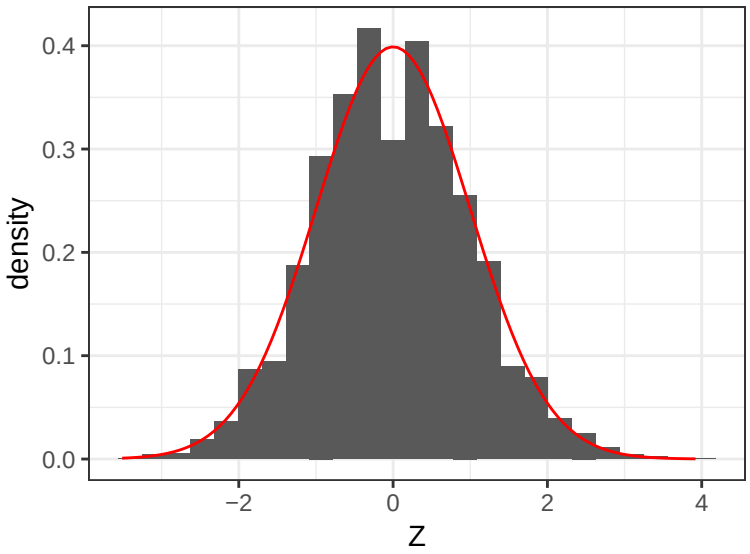
3. Construct a histogram of the quantites from question 2.

```
df.lhs <- data.frame(Z)
ggplot(df.lhs,aes(Z)) +
  geom_histogram(aes(y= stat(density)),bins=25) +
  theme_bw()
```



4. On the figure from question 3, overlay the pdf of the normal distribution corresponding to the right hand side of the expression. Hint: use stat_function(fun=dnorm).

```
ggplot(df.lhs,aes(Z)) +
  geom_histogram(aes(y= stat(density)),bins=25) +
  stat_function(fun = dnorm, args = list(mean = 0, sd = 1),col = 'red') + #add density
  theme_bw()
```

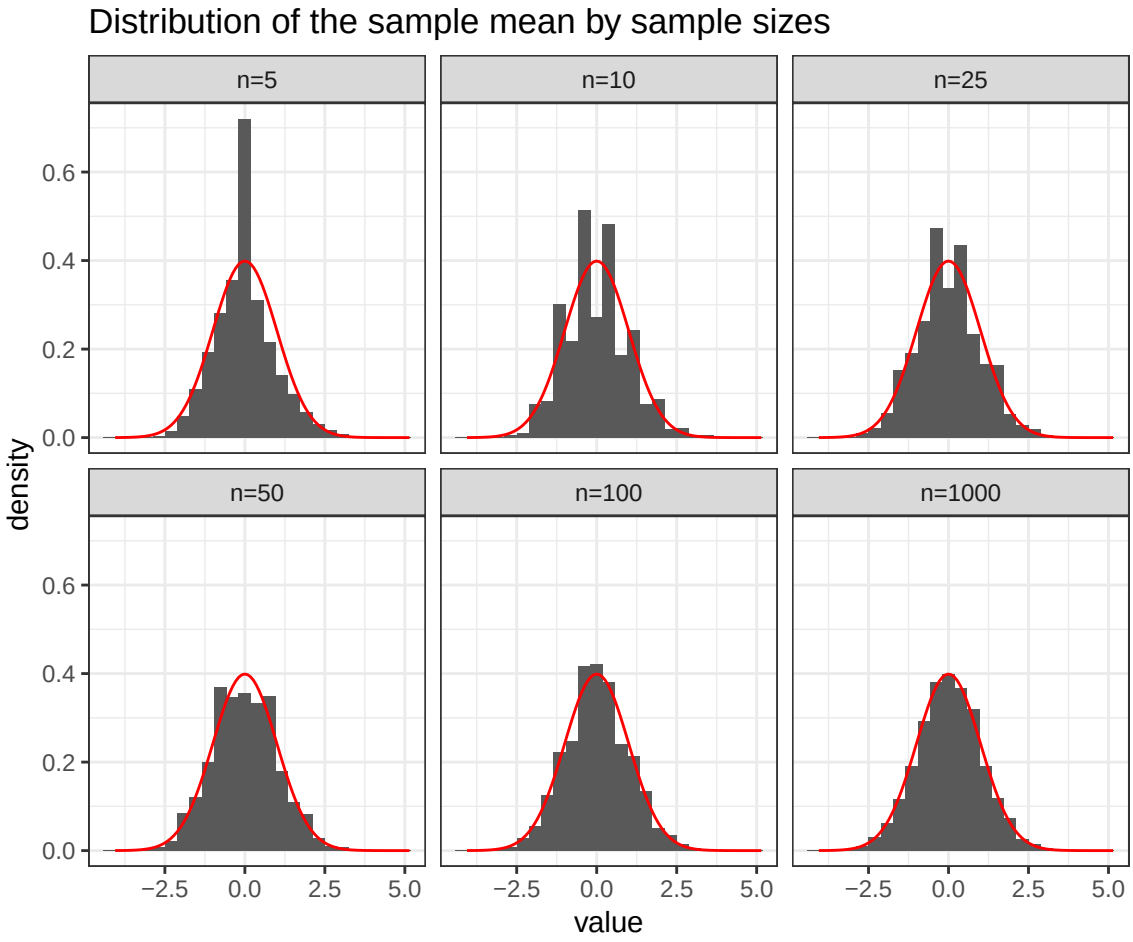


5. Now, repeat questions 2-4 using various sample sizes of $N = (5, 10, 25, 50, 100, 1000)$. You will want to collect all your results in one data frame for plotting.

```
set.seed(9876)
N <- c(5,10,25,50,100,1000)
set.seed(4)
results <- sapply(N,function(n,mu,sd=sigma) {
  sapply(1:R,function(x) {sqrt(n)*(mean(rpois(n,mu)) - mu)/sigma})}, mu=lambda)

#using a for loop
set.seed(4)
results <- matrix(NA,R,length(N))
for (n in 1:length(N)){
  for (i in 1:R){
    results[i,n] <- sqrt(N[n])*(mean(rpois(N[n],lambda)) - lambda)/sigma
  }
}
```

6. Plot the results from question 5 in ONE figure (i.e. panel of six plots) using the facet_wrap() function. Depending on how you saved your results from 5, you may also need to use the melt() function to get your data frame into the appropriate format for ggplot2-ing.

```
colnames(results) <- paste0("n=",N)
results %>% melt() -> results.df

ggplot(results.df,aes(value)) +
  geom_histogram(aes(y=stat(density)),bins=25) +
  facet_wrap(~Var2) +
  stat_function(fun = dnorm, args = list(mean = 0, sd = 1),col = 'red') +
  theme_bw() + labs(title="Distribution of the sample mean by sample sizes")
```



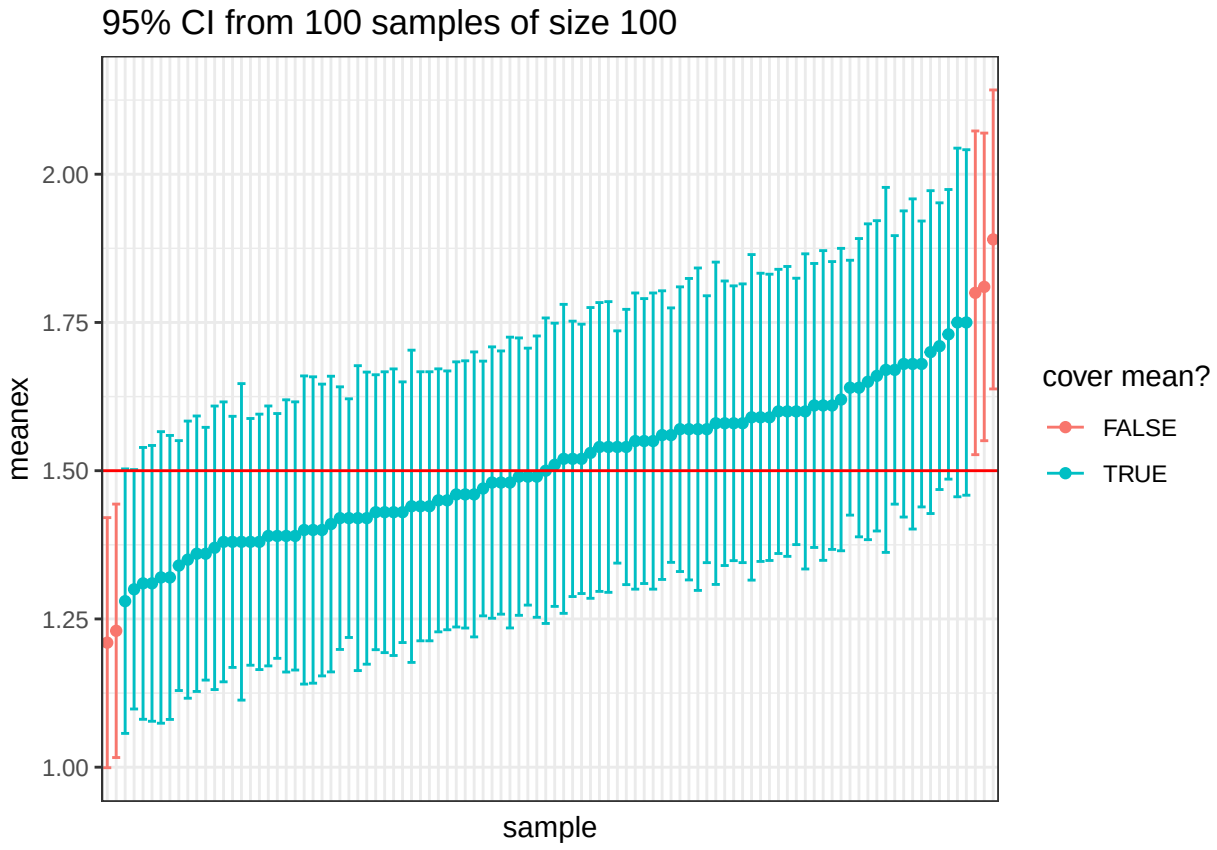
7. Let's motivate how this relates to confidence intervals for the population mean! Calculate the confidence intevals for 100 random samples of size $n = 100$ from a Poisson distribution with λ equal to 1.5. Plot the confidence intervals and include a line for the true value of λ. How often is the true mean captured? Does this match expectations?

Tip: to make it easier to see which confidence intervals cover the true mean, you may want to sort or color the intervals.

```
set.seed(98)
lambda=1.5
n=100 #size of samples
R=100 #number of random samples
xsamp <- sapply(1:R,function(x) {rpois(n,lambda)})
xsamp.df <- data.frame(xsamp)
colnames(xsamp.df) <- paste0("samp",1:R)

xsamp.df %>% melt() %>% group_by(variable) %>%
  summarise(meanex=mean(value),sd=sd(value),n=n()) %>%
  arrange(meanex) %>%
  mutate(variable = factor(variable, variable),
         ci_low = meanex-qnorm(.975)*sd/sqrt(n),
         ci_up = meanex+qnorm(.975)*sd/sqrt(n)) %>%
  mutate(`cover mean?` = (lambda > ci_low & lambda < ci_up)) -> df
```

```
## No id variables; using all as measure variables
```

```
ggplot(df,aes(y=meanex,x=variable,color=`cover mean?`)) +
  geom_point() + geom_errorbar(aes(ymin=ci_low, ymax=ci_up)) +
  geom_hline(yintercept=1.5, col="red") + theme_bw() +
  theme(axis.text.x=element_blank(),axis.ticks.x = element_blank()) +
  labs(title="95% CI from 100 samples of size 100",x="sample")
```



8. Take a moment and interpret the implications from this exercise. How should we interpret a confidence interval?

The CLT is beautiful, isn't it? This gives us the distribution for the sample mean ($\bar{X}_n$), which we can leverage to make inferences about the true population mean (μ). This is a powerful result because it holds regardless of the distribution for the underlying random variable (note that the Poisson distribution with $\lambda = 1.5$ is not normally distributed).

Great references / more practice with tidyverse

- Hadley's R for Data Science book or website
- Specifically, the ggplot2 exercises
- Tidyverse cheatsheet by DataCamp