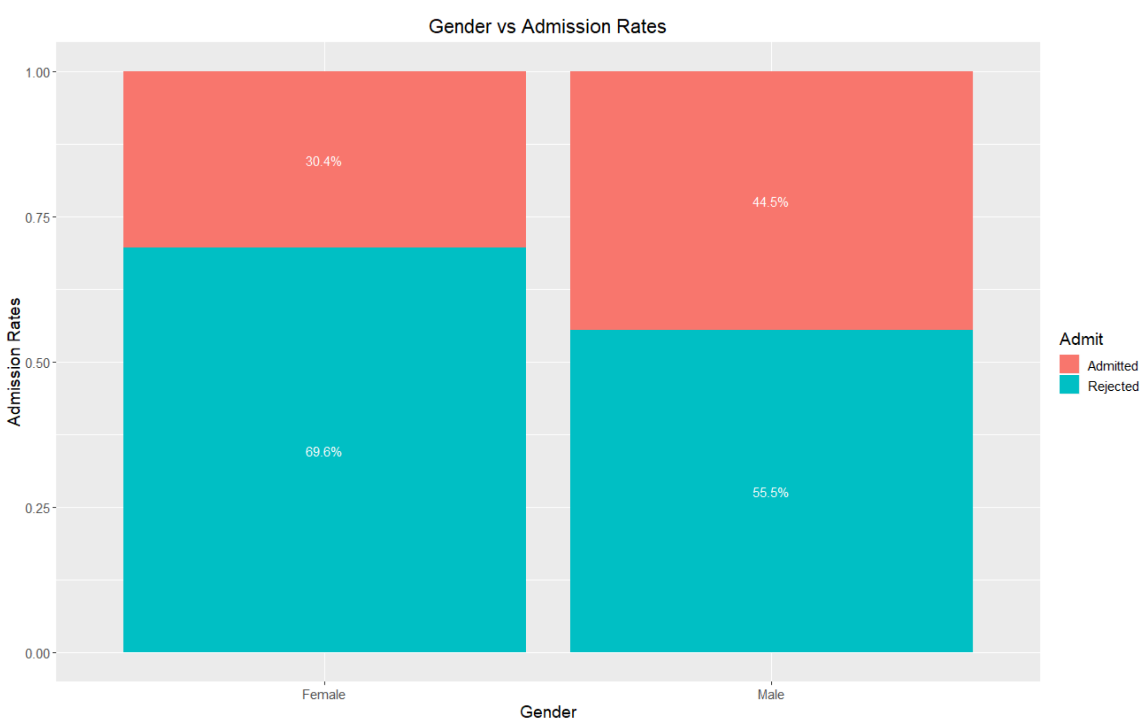
A. College admissions dataset

1. Provide a plot that shows the admission rates for men and women separately. Make sure that your plot has an interpretable title, legend, and axes labels.

```
ca = read.csv("http://vicpena.github.io/admin.csv")
ca <- ca %>% uncount(Freq)
# label added
percentdata <- ca %>% group_by(Gender) %>% count(Admit) %>%
  mutate(ratio=scales::percent(n/sum(n)))
percentdata

ggplot(ca) +
  aes(x = Gender, fill = Admit) +
  geom_bar(aes(y = (..count..)/sum(..count..)), position = "fill") +
  ylab("Admission Rates") +
  ggtitle("Gender vs Admission Rates") +
  theme(plot.title = element_text(hjust = 0.5)) +
```



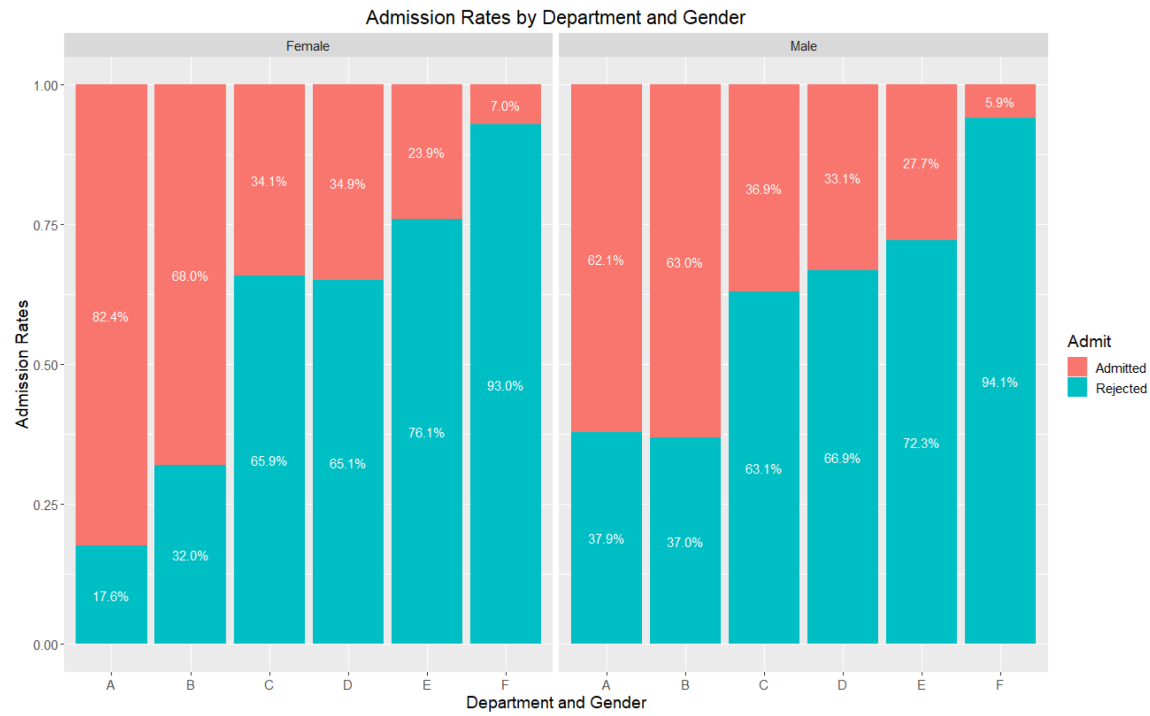
```
  geom_text(data = percentdata, aes(y=n, label=ratio), colour="white",
position=position_fill(vjust=0.5)) +
  theme(text=element_text(size=14))
```

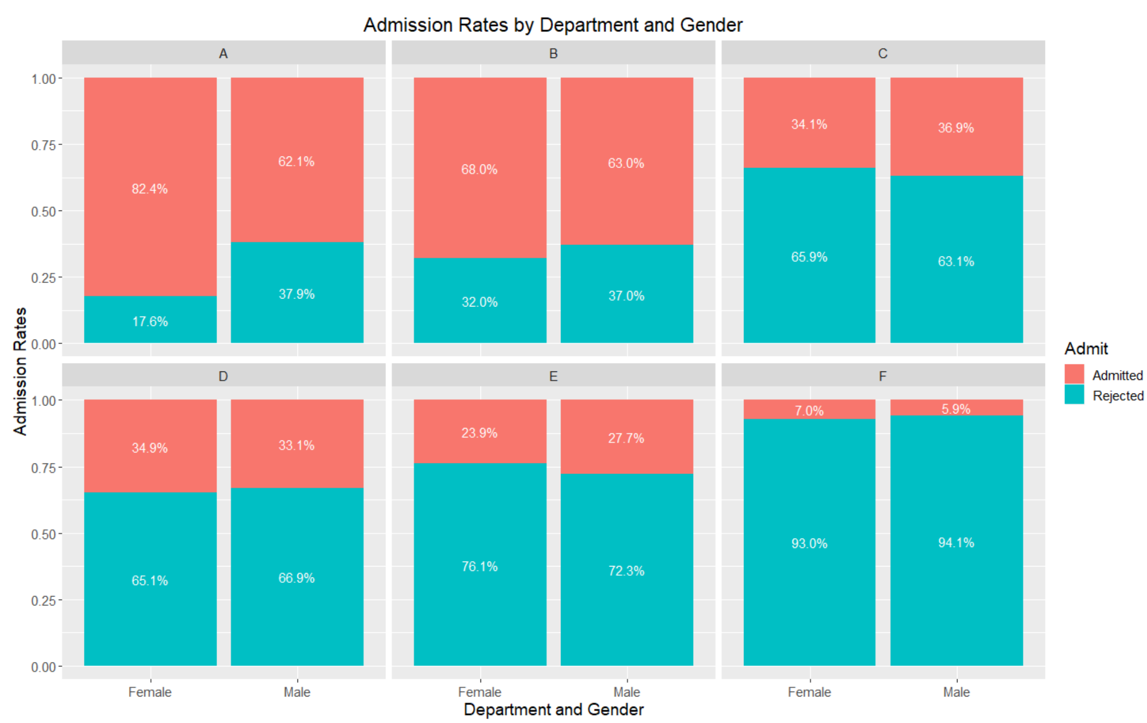
2. Provide a plot that shows admission rates by department and gender. Make sure that your plot has an interpretable title, legend, and axes labels.

```
# lebel added
percentData <- ca %>% group_by(Dept, Gender) %>% count(Admit) %>%
  mutate(ratio=scales::percent(n/sum(n)))
percentData
# 100% vertical stacked bar chart(--add text and change color schema--)
ggplot(ca) +
  aes(Dept, Gender, fill = Admit) +
  geom_bar(aes(y = (..count..)/sum(..count..)), position = "fill") +
  facet_wrap(~Gender, nrow = 1) +
  ylab("Admission Rates") +
  xlab("Department and Gender") +
  ggtitle("Admission Rates by Department and Gender") +
  geom_text(data = percentData, aes(y=n, label=ratio), colour="white",
position=position_fill(vjust=0.5)) +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(text=element_text(size=14))


ggplot(ca) +
  aes(Gender, Dept, fill = Admit) +
  geom_bar(aes(y = (..count..)/sum(..count..)), position = "fill") +
  facet_wrap(~Dept) +
  ylab("Admission Rates") +
  xlab("Department and Gender") +
  ggtitle("Admission Rates by Department and Gender") +
  geom_text(data = percentData, aes(y=n, label=ratio), colour="white",
position=position_fill(vjust=0.5)) +
```

```
theme(plot.title = element_text(hjust = 0.5)) +
theme(text=element_text(size=14))
```





3. Give a one-paragraph explanation of what you see in the plots in your own words. Make sure that your explanation isn't too technical.

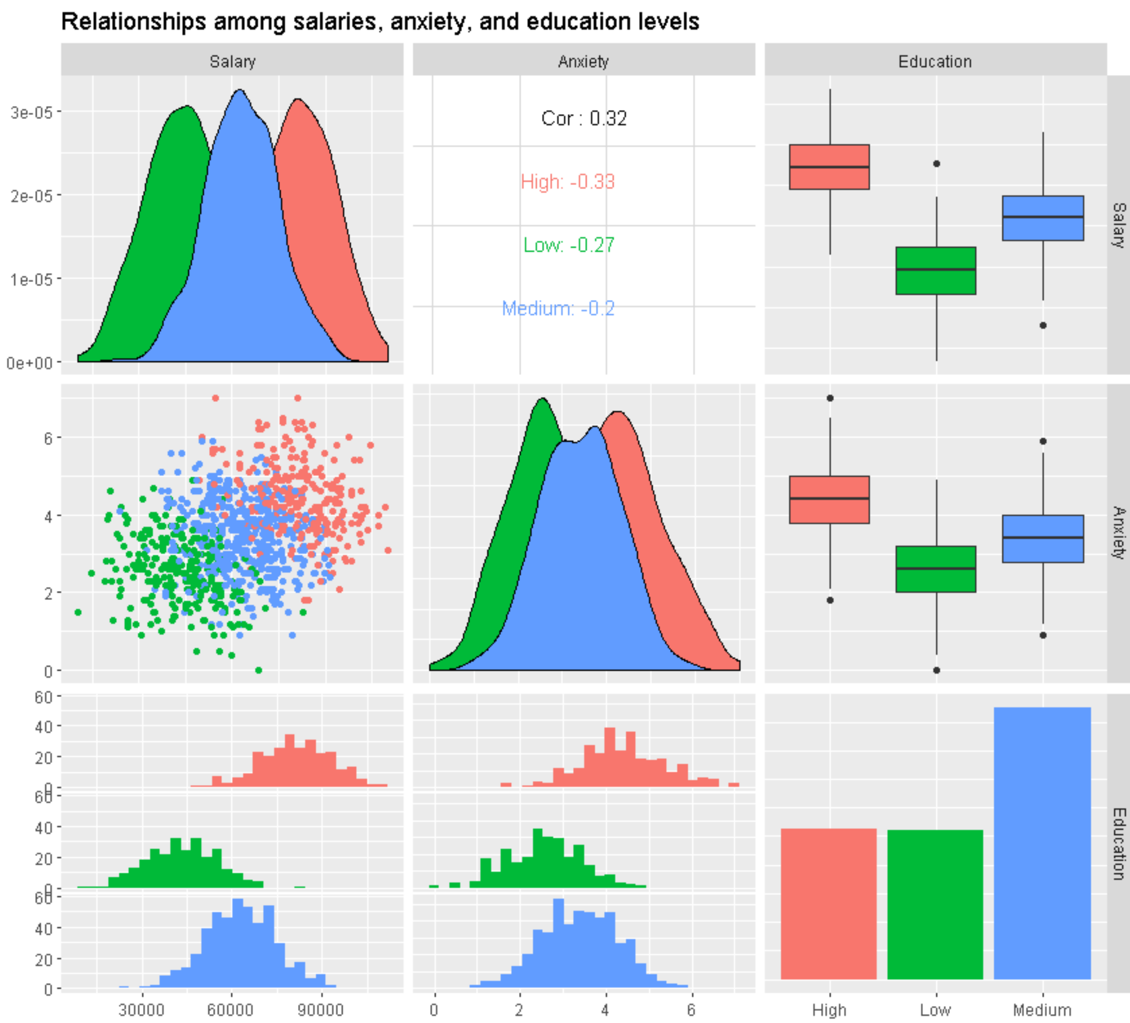
From the first plot above, we could see that college admission seems to prefer male than female, however, that's not true expect for department E. As we can see from the comparison between male and female admission rate of each department that all departments except for department E prefer female than male.

B. Salaries and anxiety

1. Create a figure that displays the relationship between salaries, anxiety, and education levels. Make sure that the labels and the title are interpretable. Interpret in detail the relationships that you see.

```
salary = read.csv("http://vicpena.github.io/sta9750/salary.csv")
```

```
ggpairs(data=salary, title="Relationships among salaries, anxiety, and education levels", mapping=ggplot2::aes(colour = Education))
```



As we can see from the ggpairs plot above that people with higher education tend to have a higher salary and anxiety level; a interesting thing to notice here is that although the relationship between anxiety and salary is negative within each education group, the overall relationship between anxiety and salary is surprisingly positive.

2. An article claims that, given any educational level, higher salaries come at the cost of higher anxiety levels. Does the figure you created in part 1 support this claim?

No. According to the ggpairs plot above, the relationship between anxiety and salary within each education group is negative.

C. Call center

A call center wants to know how many operators they should have. They're open 6 hours from Monday to Friday (both included). The number of calls they get in an hour depends on whether it's a busy day or a quiet day. If it's not a busy day, the number of calls that they get in an hour is approximately Poisson(60). If it's a busy day, the number of calls they get is Poisson(200). It is estimated that the probability that any given day is a busy day is 0.2.

Answer the following questions using simulation (i.e. random numbers) unless otherwise specified:

1. Assume that there is a very busy week, where all the days are busy days. Find the expected number of calls they get that week.

```
lambda = 5*6*200

nsim = 1e5

call = rpois(nsim, lambda = lambda)

mean(call)
```

Result:

```
> mean(call)
[1] 5999.93
```

2. [You don't need simulation for this one.] Assume that an operator can take approximately 30 calls per hour. How many operators should they get if they want to make sure that they can handle busy days well?

```
nsim = 1e5

sim = rpois(nsim, lambda = 6*200)

round(quantile(sim)/30,0)
```

Result:

```
> round(quantile(sim)/30,0)
  0%  25%  50%  75% 100% 
  35   39   40   41   45 
```

If the call center want to make sure that every customer can have their call without waiting 100%, they could hire 45 operators.

3. What is the average number of calls they get in a day?

```
busy = 6*200

quiet = 6*60


# simulate quiet and busy days

# 1 if busy, 0 if quiet

nsim = 1e5


day = rbinom(nsim, size = 1, prob = 0.2)

tab = table(day)

tab


call_busy = rpois(tab[2], lambda = busy)

call_normal = rpois(tab[1], lambda = quiet)

calls = c(call_busy, call_normal)

mean(calls)
```

Result:

```
> busy = 6*200
> quiet = 6*60
> 
> # simulate quiet and busy days
> # 1 if busy, 0 if quiet
> nsim = 1e5
> 
> day = rbinom(nsim, size = 1, prob = 0.2)
> tab = table(day)
> tab
day
    0     1 
80100 19900 
> 
> call_busy = rpois(tab[2], lambda = busy)
> call_normal = rpois(tab[1], lambda = quiet)
> calls = c(call_busy, call_normal)
> mean(calls)
```

```
[1] 527.1205
```

4. [You don't need simulation for this one. There isn't a definitive right or wrong question here. Make assumptions as needed. I'll grade you based on insight and coherence.] Given what you learned in parts 1 and 3, how many operators would you have in the call center?

```
average_sim = rpois(nsim, lambda = (0.2*(6*200) + 0.8*(6*60)))

round(quantile(average_sim)/30,0)


busy_sim = rpois(nsim, lambda = busy)

round(quantile(busy_sim)/30,0)


quiet_sim = rpois(nsim, lambda = quiet)

round(quantile(quiet_sim)/30,0)
```

Result:

```
> average_sim = rpois(nsim, lambda = (0.2*(6*200) + 0.8*(6*60)))
> round(quantile(average_sim)/30,0)
  0%  25%  50%  75% 100%
  14   17   18   18   21
>
> busy_sim = rpois(nsim, lambda = busy)
> round(quantile(busy_sim)/30,0)
  0%  25%  50%  75% 100%
  35   39   40   41   45
>
> quiet_sim = rpois(nsim, lambda = quiet)
> round(quantile(quiet_sim)/30,0)
  0%  25%  50%  75% 100%
   9   12   12   12   15
```

If the call center would like to have nice service and make sure the callers don't have to wait to get their call picked up 10% of the time when it's a busy day, then they could have 45 operators; if they just want to make sure they handle quiet days well, then they could only hire 15 operators; The company also could also hire 21 operators to make sure they handle average days well.

5. What is the average number of calls they will get in the month of December?

31 days in Dec

busy = 31*6*200

quiet = 31*6*60

```
# simulate quiet and busy days

# 1 if busy, 0 if quiet

nsim = 1e5


day = rbinom(nsim, size = 1, prob = 0.2)

tab = table(day)

tab


call_busy = rpois(tab[2], lambda = busy)

call_normal = rpois(tab[1], lambda = quiet)

calls = c(call_busy, call_normal)

mean(calls)
```

Result:

```
> mean(calls)
[1] 16429.47
```

D. Restaurant

A restaurant is open every day. On weekdays, sales are approximately normal with a mean of \$2000 and a standard deviation of \$500. On the weekends, sales are approximately normal with a mean of \$3000 and a standard deviation of \$700. The rent costs \$2500 weekly, labor costs \$4500 weekly, food costs \$4500 weekly, and other expenses amount to \$2500 weekly.

Use simulation (unless otherwise specified) to answer the following questions:

1. In any given week, what is the probability that the restaurant is making money?

```
# weekday sales ~ Normal(2000, sd = 500)

# weekend sales ~ Normal(3000, sd = 700)

# weekly sales ~ Normal(5*2000+2*3000, sd = sqrt(5*500^2+2*700^2))

weeklycosts = 2500+4500+4500+2500


# P(weeklysales > weeklycosts)

pnorm(weeklycosts,

      mean = 5*2000+2*3000,
```

```
      sd = sqrt(5*500^2+2*700^2),
      lower.tail = FALSE)
```

Result:

```
[1] 0.9097633
```

2. What is the probability that they lose money in December?

```
# week days = 31-9 ~ Normal(2000, sd = 500)
# weekend days = 9 ~ Normal(3000, sd = 700)
# dist of monthly sales in December
# Nor((31-9)*2000+9*3000, sd = sqrt((31-9)*500^2+9*700^2))

daylycosts = weeklycosts/7
monthlycosts = 31*daylycosts
monthlycosts

# P( monthlysales < monthlycosts)
pnorm(monthlycosts,
      mean = (31-9)*2000+9*3000,
      sd = sqrt((31-9)*500^2+9*700^2))
```

Result:

```
[1] 0.002125268
```

3. What is the yearly expected profit of the business? [Assuming all costs remain constant.]

yearlycosts = weeklycosts*52

expyearlysales = (5*2000+2*3000)*52

expyearlysales-yearlycosts

Result:

```
 [1] 104000
```

4. [For this question, you don't have to use simulation to support all your claims, if you want. There isn't a definitive right or wrong question here. Make assumptions as needed. I'll grade you based on insight and coherence.] The manager is thinking about running a marketing

campaign which costs $10,000. The probability that the marketing campaign is successful is 80%. If it is successful, average sales on weekdays and weekends will increase by 15% (the standard deviation will stay the same). Would you recommend running the marketing campaign? Why or why not?

```
yearlycosts = weeklycosts*52

expyearlysales = (5*2000+2*3000)*52


# 0 is fail, 1 is success

outcome = function(){

  result = sample(x = c(0,1),

         size = 1, prob = c(0.2,0.8), replace = FALSE)

  result

}


random_simulation = function(marketing_campaign){

  if (marketing_campaign == 1){

    netprofit = expyearlysales*0.15 - 10000

  }

  if (marketing_campaign == 0){

    netprofit = -10000

  }

  netprofit

}


totalearning = 0


for (i in 1:1e5){

  result = outcome()

  earning = random_simulation(result)

  totalearning = totalearning + earning
```

```
}

avgearning = totalearning/1e5

avgearning
```

Result:

```
> avgearning
[1] 89841.25
```