STA 380 Part 2: Exercises

Aidan Cremins, Peyton Lewis, Joe Morris, Amrit Sandhu

2022-07-29

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
library(forcats)
library(reshape2)
library(knitr)
```

Probability Practice

Part a.

P(Y) = 0.65 P(N) = 0.35 P(RC) = 0.3 P(TC) = 0.7 (P(RC)-1) P(Y|RC) = 0.5 P(N|RC) = 0.5

These probabilities are summarized in the table below:

We're looking for P(Y|TC) so we can use the rule of total probability:

P(Y) = P(Y, TC) + P(Y, RC) = P(TC) * P(Y|TC) + P(RC) * P(Y|RC)

We know all of these inputs to the equation except for P(Y|TC), so we want to solve for that unknown.

0.65 = 0.7 * P(Y|TC) + 0.3 * 0.5

From the above equation, we find that P(Y|TC) $\approx$ 0.714286. This means that truthful clickers answer yes to the question about 71.43% of the time.

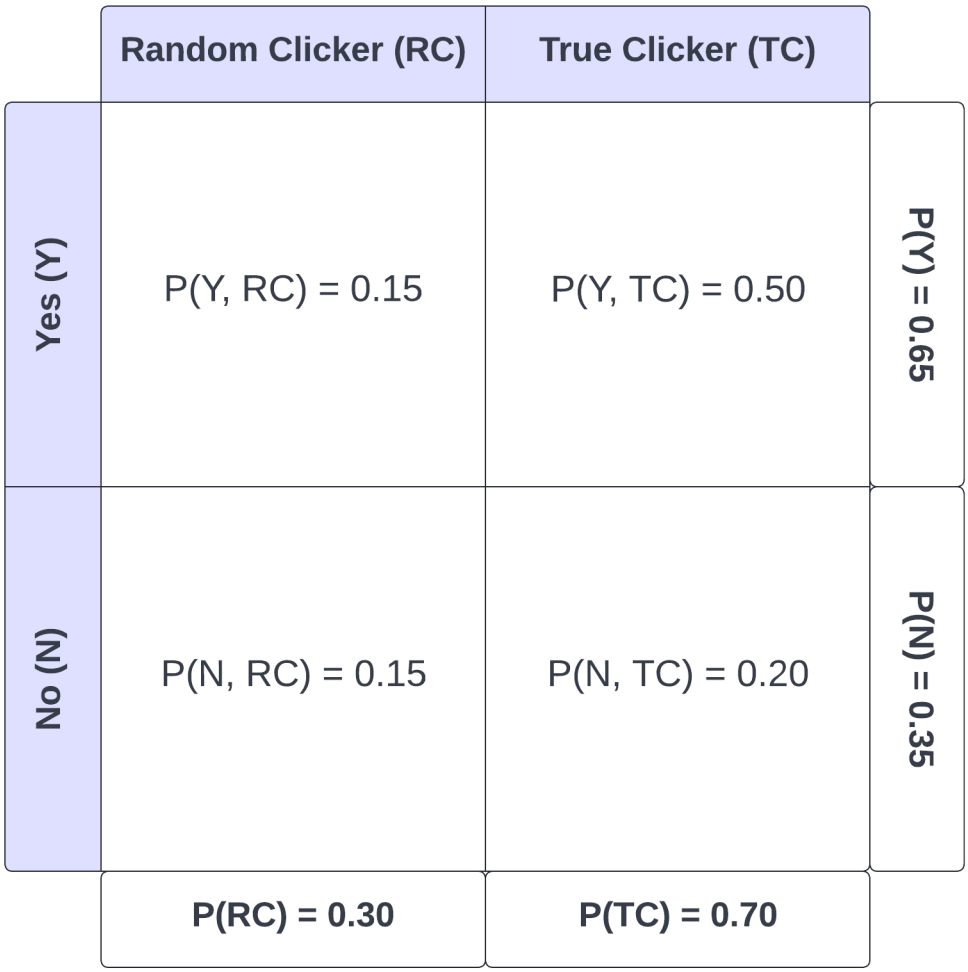

Random Clicker (RC)
True Clicker (TC)
Yes (Y)
P(Y, RC) = 0.15
P(Y, TC) = 0.50
P(Y) = 0.65
No (N)
P(N, RC) = 0.15
P(N, TC) = 0.20
P(N) = 0.35
P(RC) = 0.30
P(TC) = 0.70

Figure 1: alt

Part b.

P(Disease) = 0.000025 P(No Disease) = 0.999975 (1-0.000025) P(Positive|Disease) = .993 P(Negative|No Disease) = 0.9999

The probabilities above are summarized in the tree diagram below:

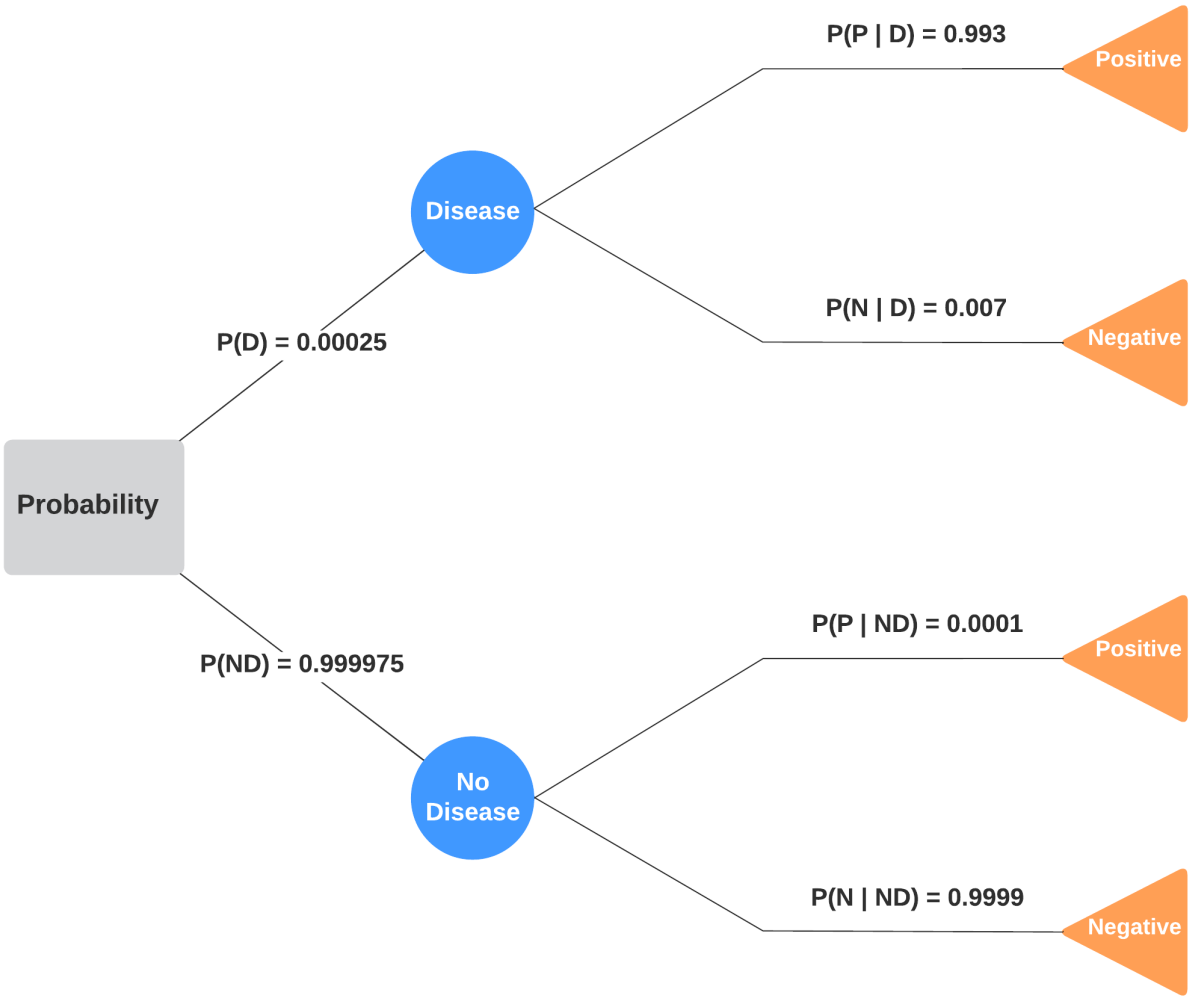


Figure 2: alt

We're looking for P(Disease|Positive) so we can use Baye's Law:

$$\frac{P(Disease)*P(Positive|Disease)}{P(Disease)*P(Positive|Disease)+P(NoDisease)*P(Positive|NoDisease)}$$

We have almost all of the inputs that we need, however, we're missing P(Positive|No Disease). These are false positives. We can find the missing probability by taking 1 - true negatives, or 1 - 0.9999 to get P(Positive|No Disease) as 0.0001. Now we can solve for P(Disease|Positive).

$\frac{0.000025*0.993}{0.000025*0.993+0.999975*0.0001} \approx .198882$. Thus, if someone tests positive, they have about a 19.89% chance of actually having the disease.

Wrangling the Billboard Top 100

```
billboard = read.csv("data/billboard.csv")
```

#Need a caption - probably something about how most are recent songs

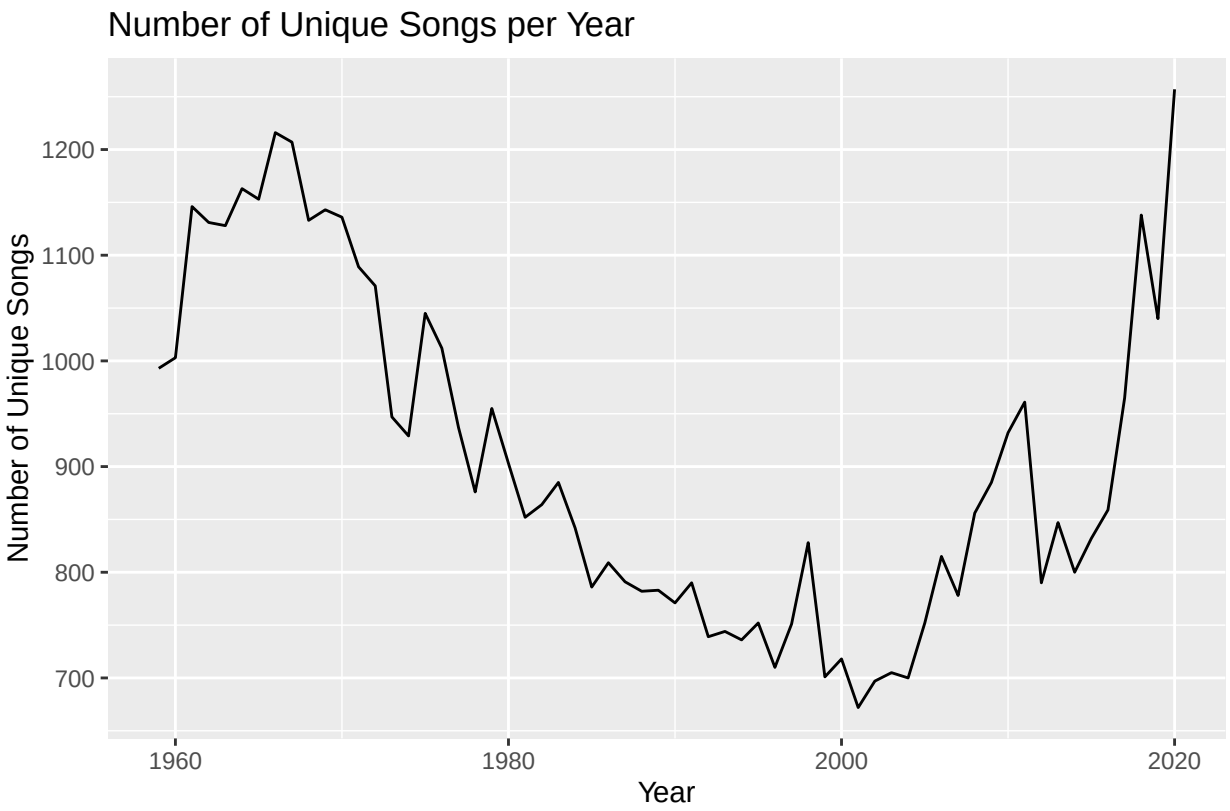
Part a.

The table below shows the top 10 longest lasting songs on the Billboard 100. It reveals that a majority of these long-lasting songs are more recent songs.

Part b.

```
musical_diversity = billboard %>%
  filter(year != 1958 & year != 2021) %>%
  group_by(year) %>%
  summarize(unique_songs_per_year = length(unique(c(performer,song))))

ggplot(musical_diversity) + geom_line(aes(x = year, y = unique_songs_per_year)) + xlab("Year") + ylab("
```



). Since then, the trend has been increasing and in the last year of data (2020), there were about 1250 unique songs.

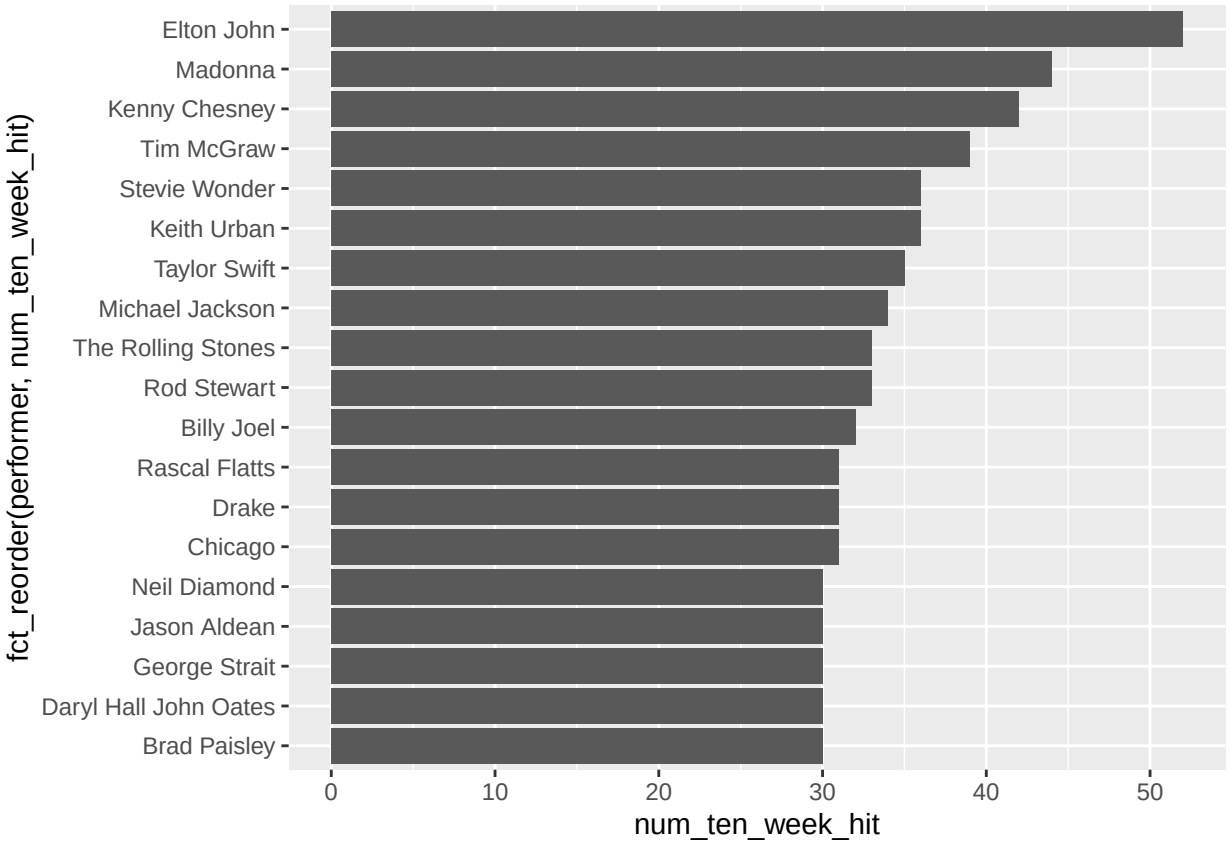
Part c.

```
ten_week_hit_songs <- billboard %>%
  group_by(performer,song) %>%
  summarize(ten_week_hit = ifelse(n()>=10,"Yes","No")) %>%
  filter(ten_week_hit == "Yes")
```

```
## `summarise()` has grouped output by 'performer'. You can override using the
## `.groups` argument.
```

```
top_artists <- ten_week_hit_songs %>%
  group_by(performer) %>%
  summarize(num_ten_week_hit = n()) %>%
  filter(num_ten_week_hit>=30)

ggplot(top_artists) + geom_bar(aes(x = fct_reorder(performer,num_ten_week_hit), y = num_ten_week_hit),s
```

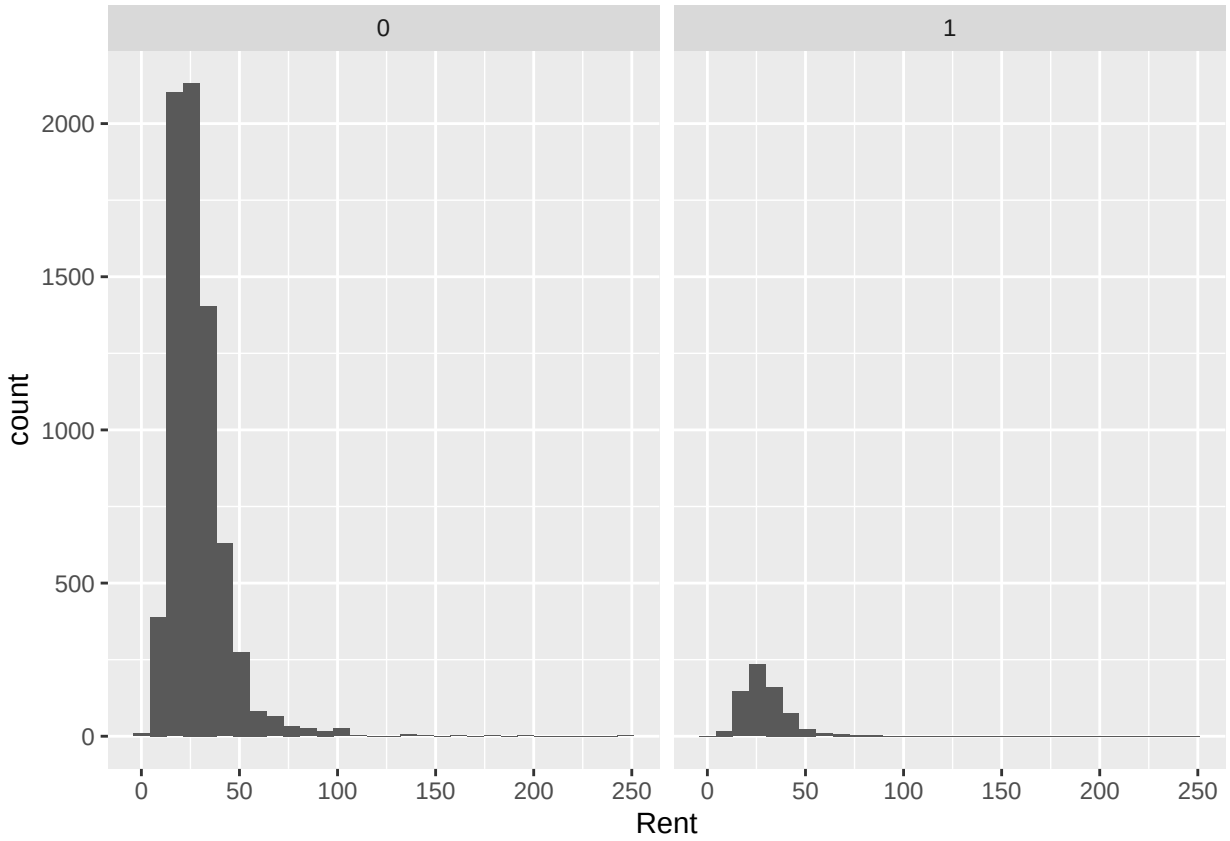


#Visual story telling part 1: green buildings

```
green_buildings = read.csv("data/greenbuildings.csv")
```

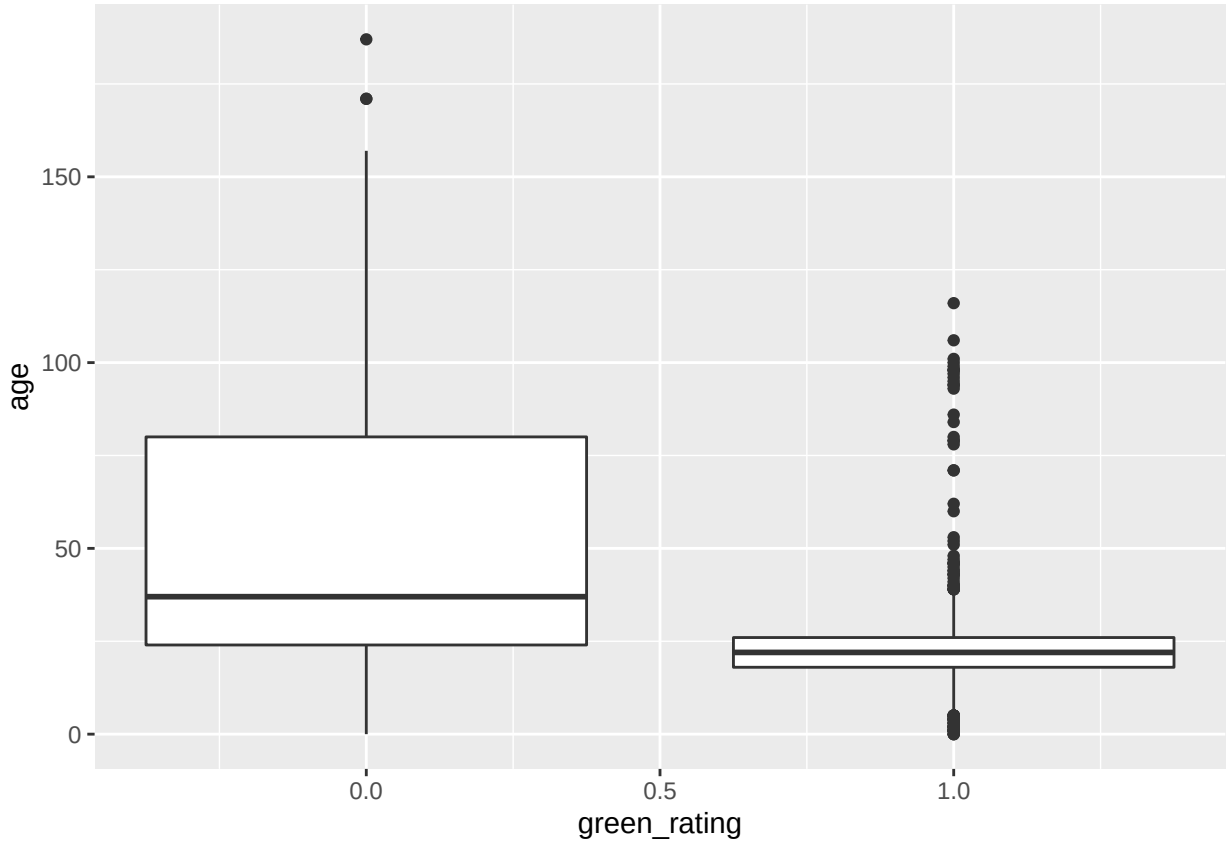
```
ggplot(green_buildings, aes(x = Rent)) + geom_histogram() + facet_grid(.~green_rating)
```

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

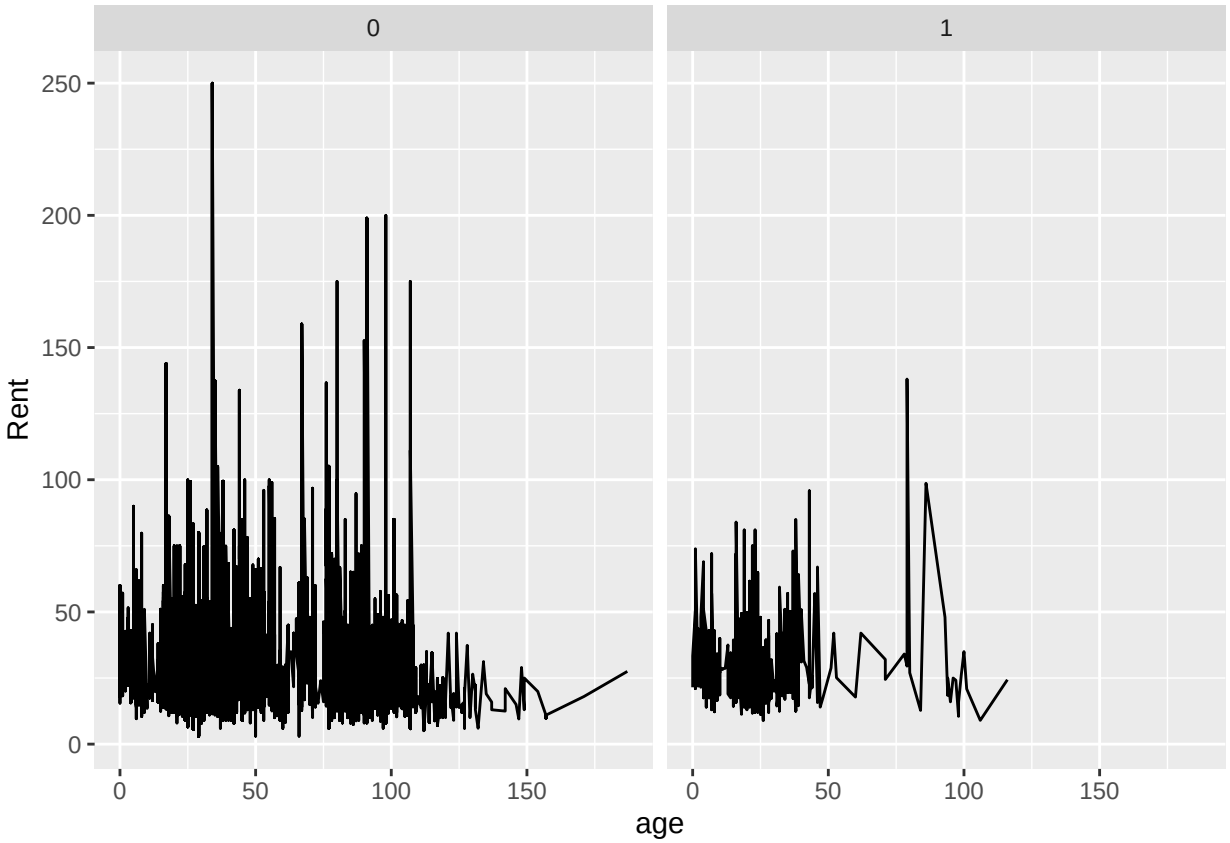


Green buildings tend to be newer, "newness" could justify higher rents

```
ggplot(green_buildings, aes(x = green_rating, y = age ,group = green_rating)) + geom_boxplot()
```



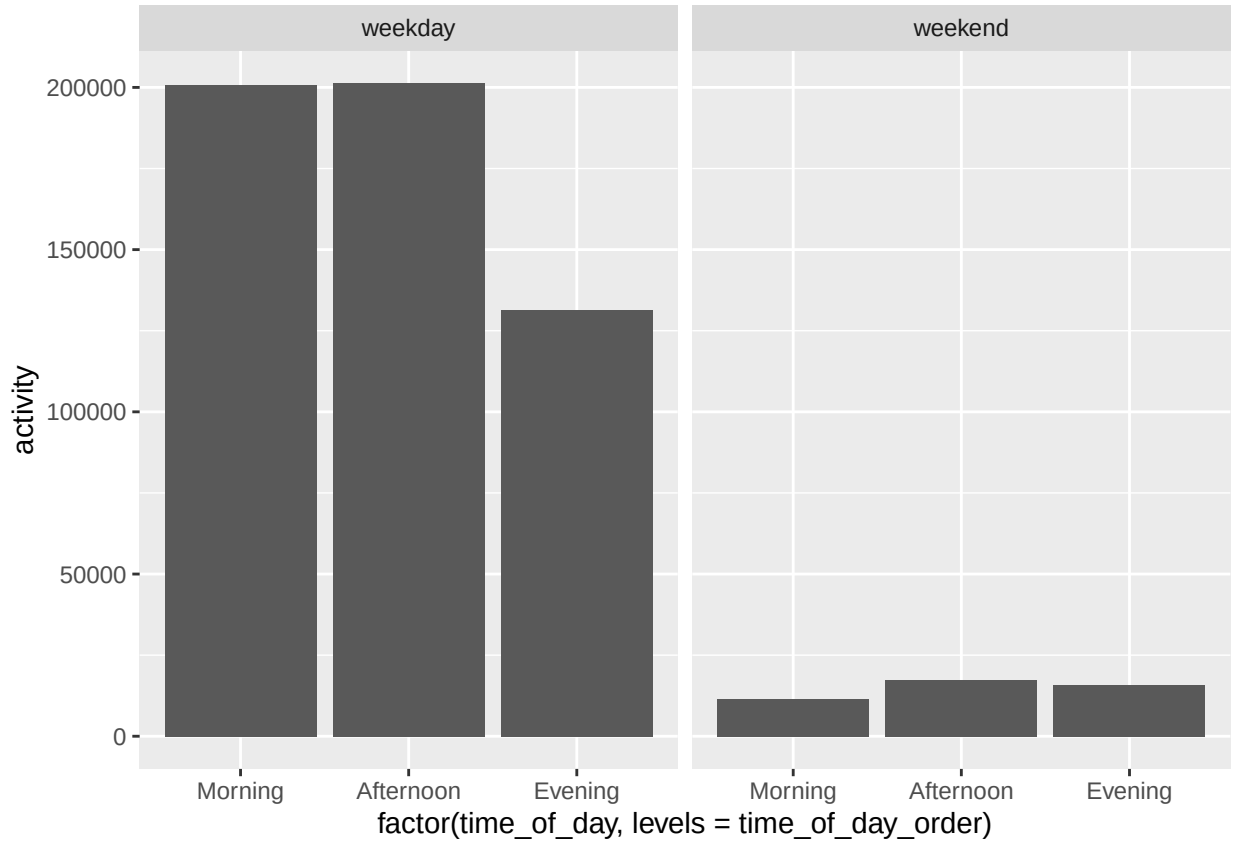

```
ggplot(green_buildings, aes(x = age, y = Rent)) + geom_line() + facet_grid(. ~ green_rating)
```



Visual story telling part 2: Cap Metro data

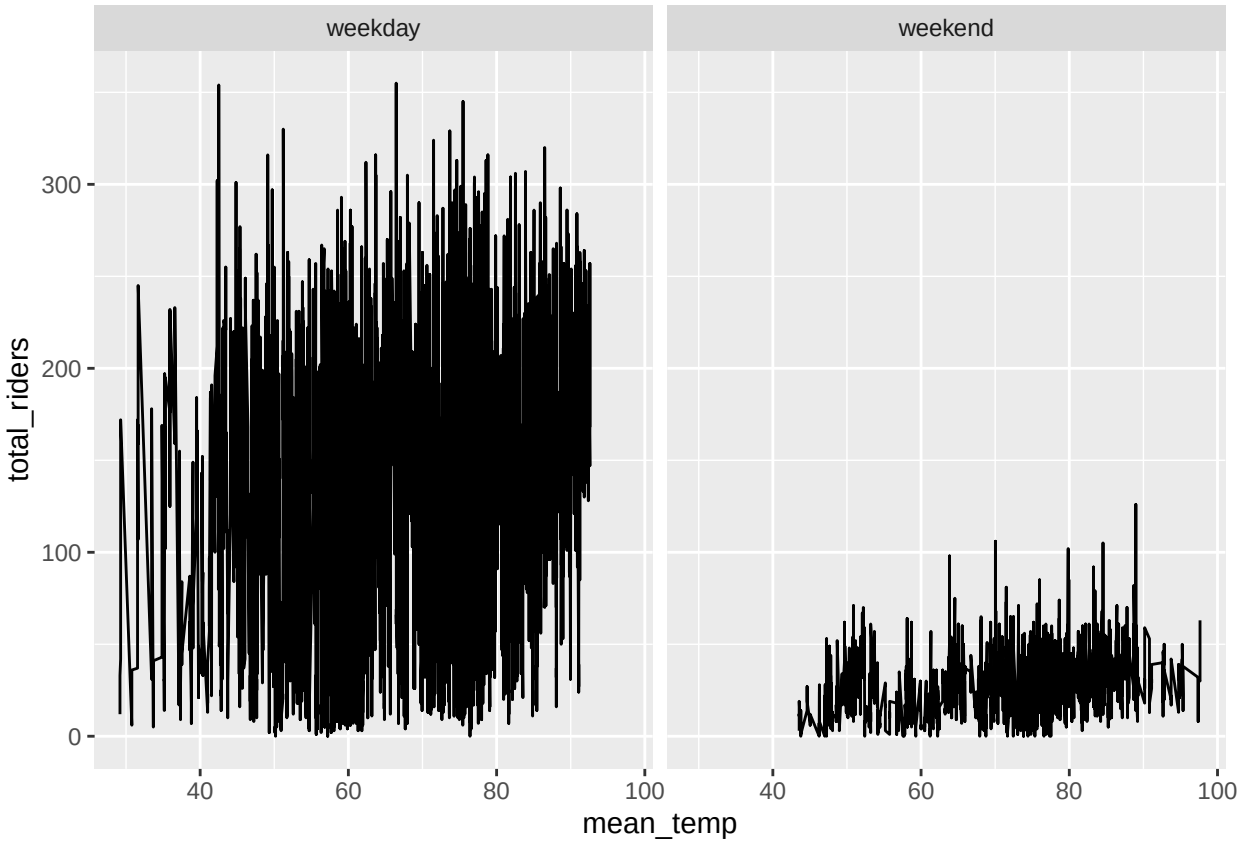
```
cap_metro <- read.csv("data/capmetro_UT.csv")
```

```
cap_metro$time_of_day = ifelse(cap_metro$hour_of_day %in% c(6,7,8,9,10,11),"Morning",ifelse(cap_metro$h
time_of_day_order <- c("Morning","Afternoon","Evening")
cap_metro$activity = cap_metro$boarding + cap_metro$alighting
ggplot(cap_metro, aes(x = factor(time_of_day,levels=time_of_day_order), y = activity)) + geom_bar(stat=
```



Activity seems to slightly increase as temperature increases, adjusted for the difference in ridership between weekdays and weekends

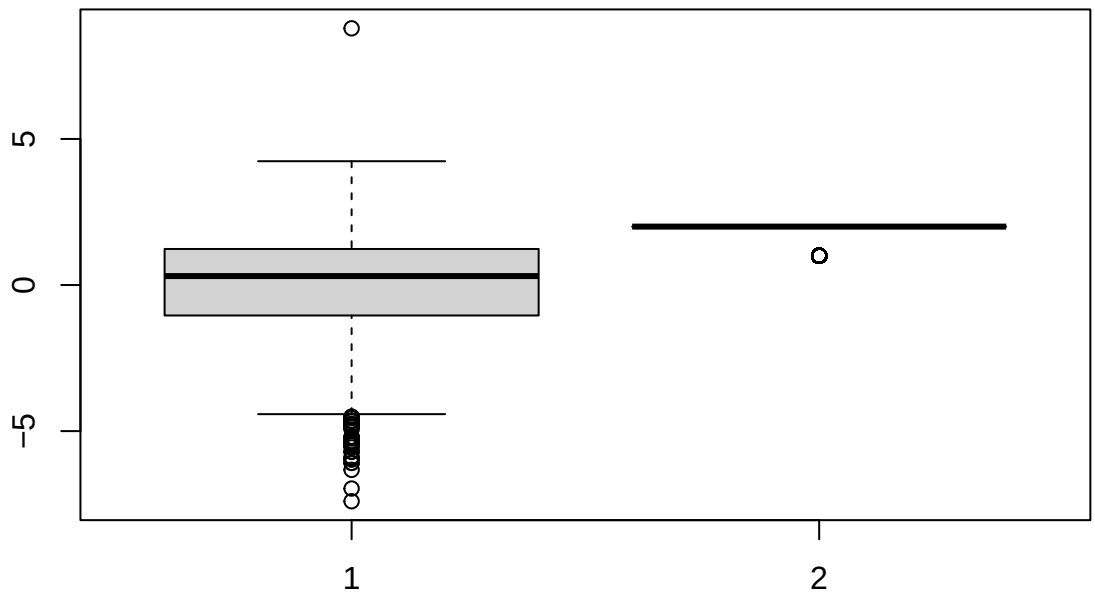
```
riders_temp = cap_metro %>%
  group_by(timestamp) %>%
  summarize(total_riders = sum(activity), mean_temp = mean(temperature), weekend = weekend)
ggplot(riders_temp, aes(x = mean_temp, y = total_riders)) + geom_line()+facet_grid(.~weekend)
```



Clustering and PCA

```
wine <- read.csv("data/wine.csv")
```

```
set.seed(1)
wine_quant <- wine[,! names(wine) %in% c("color","quality")]
wine_pca = prcomp(wine_quant, rank=10, scale=TRUE)
boxplot(wine_pca$x[,1],as.factor(wine$color))
```



Cluster 1 is mostly red wines, whereas Cluster 2 is mostly white wines. Even just making two clusters distinguishes between the two wine colors very well.

```
set.seed(1)
library(knitr)
wine_quant_scaled <- scale(wine_quant)
wine_clusters <- kmeans(wine_quant_scaled, centers=2, nstart=50)
table(wine_clusters$cluster,wine$color)
```

```
##
##      red white
##   1 1575    68
##   2   24  4830
```

While the 2 clusters separated out the two wine colors well, they don't seem to distinguish between wine quality because the median quality is essentially the same for both clusters. Even if we increase the number of clusters pretty dramatically up to 10, there still doesn't appear to be major quality differences between the boxplots.

```
wine$cluster = as.factor(wine_clusters$cluster)
ggplot(wine, aes(x = cluster, y = quality)) + geom_boxplot()
```