For the green building problem, we wanted to explore and calculate how much money we can save on energy and utility bills. Because green building should be energy efficient and we will save more money (= profit, basically)

```r
library(mosaic)
```

```
## Loading required package: dplyr
```

```
##
## 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
```

```
## Loading required package: lattice
```

```
## Loading required package: ggformula
```

```
## Loading required package: ggplot2
```

```
## Loading required package: ggstance
```

```
##
## Attaching package: 'ggstance'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     geom_errorbarh, GeomErrorbarh
```

```
##
## New to ggformula?  Try the tutorials:
##  learnr::run_tutorial("introduction", package = "ggformula")
##  learnr::run_tutorial("refining", package = "ggformula")
```

```
## Loading required package: mosaicData
```

```
## Loading required package: Matrix
```

```
## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2
```

```
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected
by this.
##
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.
```

```
##
## Attaching package: 'mosaic'
```

```
## The following object is masked from 'package:Matrix':
##
##     mean
```

```
## The following object is masked from 'package:ggplot2':
##
##     stat
```

```
## The following objects are masked from 'package:dplyr':
##
##     count, do, tally
```

```
## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median,
##     prop.test, quantile, sd, t.test, var
```

```
## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```
library(readr)
library(tidyverse)
```

```
## ── Attaching packages ───────────────────────────────────────────────────────────────
───────────────────────────────────────────────────── tidyverse 1.2.1 ──
```

```
## ✔ tibble  2.1.3     ✔ purrr   0.3.2
## ✔ tidyr   0.8.3     ✔ stringr 1.4.0
## ✔ tibble  2.1.3     ✔ forcats 0.4.0
```

```
## ── Conflicts ──────────────────────────────────────────────────────────
──────────────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ mosaic::count()            masks dplyr::count()
## ✖ purrr::cross()             masks mosaic::cross()
## ✖ mosaic::do()               masks dplyr::do()
## ✖ tidyr::expand()            masks Matrix::expand()
## ✖ dplyr::filter()            masks stats::filter()
## ✖ ggstance::geom_errorbarh() masks ggplot2::geom_errorbarh()
## ✖ dplyr::lag()               masks stats::lag()
## ✖ mosaic::stat()             masks ggplot2::stat()
## ✖ mosaic::tally()            masks dplyr::tally()
```

```
green = read_csv("greenbuildings.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

```
head(green)
```

```
## # A tibble: 6 x 23
##   CS_PropertyID cluster   size empl_gr  Rent leasing_rate stories   age
##           <dbl>   <dbl>  <dbl>   <dbl> <dbl>        <dbl>   <dbl> <dbl>
## 1        379105       1 260300    2.22  38.6         91.4      14    16
## 2        122151       1  67861    2.22  28.6         87.1       5    27
## 3        379839       1 164848    2.22  33.3         88.9      13    36
## 4         94614       1  93372    2.22  35           97.0      13    46
## 5        379285       1 174307    2.22  40.7         96.6      16     5
## 6         94765       1 231633    2.22  43.2         92.7      14    20
## # … with 15 more variables: renovated <dbl>, class_a <dbl>, class_b <dbl>,
## #   LEED <dbl>, Energystar <dbl>, green_rating <dbl>, net <dbl>,
## #   amenities <dbl>, cd_total_07 <dbl>, hd_total07 <dbl>,
## #   total_dd_07 <dbl>, Precipitation <dbl>, Gas_Costs <dbl>,
## #   Electricity_Costs <dbl>, cluster_rent <dbl>
```

```
cluster<- green %>% group_by(cluster)
```

We have ordered the data by their cluster, and we analyze the hot days & cold days total to figure out the demand for energy distribution across the clusters.

```
summary(green$hd_total07)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0    1419    2739    3432    4796    7200
```

```
summary(green$cd_total_07)
```

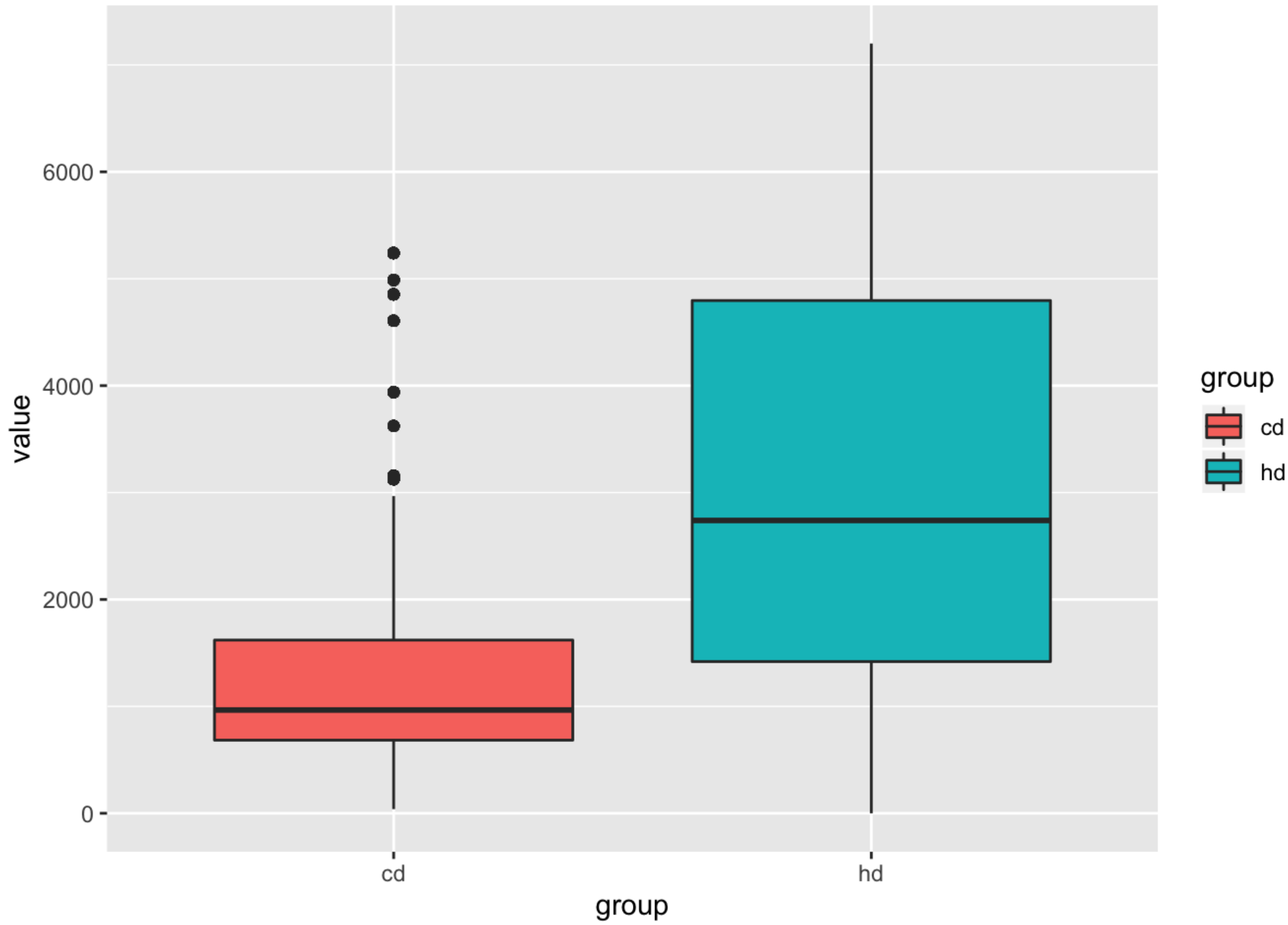
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      39     684     966    1229    1620    5240
```

We decide to only exam the clusters that have similar weather to Austin, so our conclusion would be more relatable and convincing. We take out the data with “net=1” because residents pay their own utility bills and as building owners won’t save much.

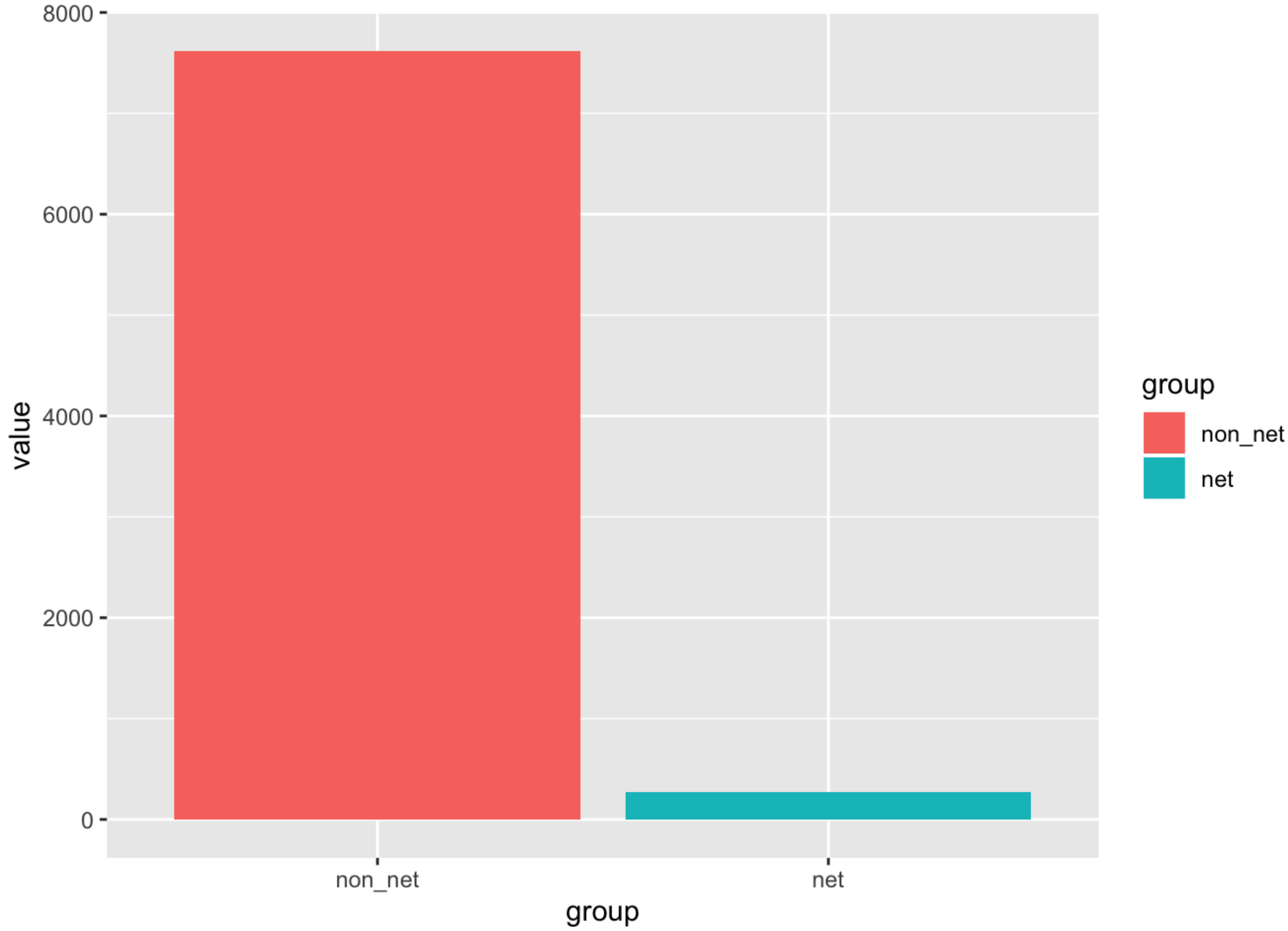
```
green_zero = green %>% filter(net == 0 )
green_one = green %>% filter(net == 1 )

a = data.frame(group ="cd",value= green_zero$cd_total_07)
b = data.frame(group = "hd", value= green_zero$hd_total07)
plot.data = rbind(a, b)
plot.data %>% group_by(group)
```

```
## # A tibble: 15,240 x 2
## # Groups:   group [2]
##    group value
##    <fct> <dbl>
##  1 cd     4988
##  2 cd     4988
##  3 cd     4988
##  4 cd     4988
##  5 cd     4988
##  6 cd     4988
##  7 cd     2746
##  8 cd     2746
##  9 cd     2746
## 10 cd     2746
## # … with 15,230 more rows
```

```
ggplot(plot.data, aes(x=group, y=value, fill=group)) +
 geom_boxplot()
```




```
f = data.frame(group = "non_net", value = nrow(green_zero))
d = data.frame(group = 'net', value = nrow(green_one))
c = rbind(f,d)
ggplot(c, aes(x=group, y=value, fill = group))+
  geom_col()
```




To define the “Austin weather”, we take heatdays > median and colddays <median. Next we only gona work with cluster that matches this condition.

```
one_q = quantile(green_zero$cd_total_07, .5)

three_q= quantile(green_zero$hd_total07, .5)

green_filter = green_zero  %>%
  filter(green_zero$cd_total_07 < one_q)

green_filt = green_filter %>%
  filter(green_filter$hd_total07 > three_q)

print(green_filt)
```

```
## # A tibble: 788 x 23
##    CS_PropertyID cluster   size empl_gr  Rent leasing_rate stories   age
##            <dbl>   <dbl>  <dbl>   <dbl> <dbl>        <dbl>   <dbl> <dbl>
##  1       1212036      20 385469      NA  18.0         99.5      14    20
##  2       1211872      20 127982      NA  15           82.3      18    86
##  3       1212027      20  63150      NA  15           66.2       6    85
##  4       1216881      20  98725      NA  15.5         85.8       6    97
##  5       1211896      20   3000      NA  16           50         4   127
##  6       1216662      20  13100      NA  16           11.0       1    45
##  7       1211247      20 850000      NA  16.5         95.9      11    90
##  8       1211853      20  70377      NA  18           75.4       6    45
##  9       1215535      20 300000      NA  18           63.4      27    40
## 10       1216862      20 309686      NA  18.6         86.1       8    81
## # … with 778 more rows, and 15 more variables: renovated <dbl>,
## #   class_a <dbl>, class_b <dbl>, LEED <dbl>, Energystar <dbl>,
## #   green_rating <dbl>, net <dbl>, amenities <dbl>, cd_total_07 <dbl>,
## #   hd_total07 <dbl>, total_dd_07 <dbl>, Precipitation <dbl>,
## #   Gas_Costs <dbl>, Electricity_Costs <dbl>, cluster_rent <dbl>
```

```
green_filt%>% group_by(cluster)%>%summarize(count = n())
```

```
## # A tibble: 52 x 2
##    cluster count
##      <dbl> <int>
##  1      20    22
##  2      52    19
##  3      58     4
##  4      81    14
##  5      87    35
##  6      93     5
##  7     142     1
##  8     164     4
##  9     167     9
## 10     174     4
## # … with 42 more rows
```

Then we tried the exact same analysis done by the immature “data scientist” did in the problem, just on our selected dataset.

```
df = green_filt %>% filter(green_filt$leasing_rate > 10)
df_green= df %>% filter(green_rating == 1)
df_non_green= df %>% filter(green_rating != 1)

summary(df_green$Rent)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   14.00   17.97   24.00   27.51   36.36   55.94
```

```
summary(df_non_green$Rent)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5.82   18.27   23.89   24.49   29.55   75.00
```

```
green = median(df_green$Rent)
non_green = median(df_non_green$Rent)
new_rent_dif = green- non_green
amt_saved = new_rent_dif * 250000
amt_saved
```

```
## [1] 27500
```

We found out that in clusters that have similar weathcer as Austin, the rent median difference isn't that great bewteen green and non-green buildings. So there is a way smaller “extra revenue” we gonna make and so far ($27,500, comparing to 650,000 computed by the guy), it seems we will recuperate the green building costs in way longer than the 7 years time period he calculated. So bad deal?

We continue to explore the revenue generated by energy savings.

```
df_green
```

```
## # A tibble: 45 x 23
##    CS_PropertyID cluster   size empl_gr  Rent leasing_rate stories   age
##            <dbl>   <dbl>  <dbl>   <dbl> <dbl>        <dbl>   <dbl> <dbl>
##  1       1212036      20 385469   NA     18.0         99.5      14    20
##  2        717047      52 315133    3.39  25.2         81.2      20    25
##  3       1258500      58  60185    0.47  16           94.5       3    27
##  4        469978      81 408459    2.44  37.9         92.6      24    24
##  5         42051      87  56303    2.44  24.2         89.7       5    97
##  6        804325      93 239250    2.44  37.5         97.3      14    13
##  7       1271164     164  98567    0.47  14           98.2       3     5
##  8       1376269     167 149426    4.02  14.5         93.9      14    39
##  9        717026     174  30000    3.39  15.9         96.1       3    98
## 10        102132     175 215000    0.26  16           88.4       6    95
## # … with 35 more rows, and 15 more variables: renovated <dbl>,
## #   class_a <dbl>, class_b <dbl>, LEED <dbl>, Energystar <dbl>,
## #   green_rating <dbl>, net <dbl>, amenities <dbl>, cd_total_07 <dbl>,
## #   hd_total07 <dbl>, total_dd_07 <dbl>, Precipitation <dbl>,
## #   Gas_Costs <dbl>, Electricity_Costs <dbl>, cluster_rent <dbl>
```

```
df_green$gas = df_green$cd_total_07 * df_green$Gas_Costs
df_green$elec = df_green$hd_total07 * df_green$Electricity_Costs

df_green$total_saved = (df_green$gas + df_green$elec) * 0.25
df_green$total_cost = (df_green$gas + df_green$elec) * 0.75
df_green
```

```
## # A tibble: 45 x 27
##    CS_PropertyID cluster   size empl_gr  Rent leasing_rate stories   age
##            <dbl>   <dbl>  <dbl>   <dbl> <dbl>        <dbl>   <dbl> <dbl>
##  1       1212036      20 385469   NA     18.0         99.5      14    20
##  2        717047      52 315133    3.39  25.2         81.2      20    25
##  3       1258500      58  60185    0.47  16           94.5       3    27
##  4        469978      81 408459    2.44  37.9         92.6      24    24
##  5         42051      87  56303    2.44  24.2         89.7       5    97
##  6        804325      93 239250    2.44  37.5         97.3      14    13
##  7       1271164     164  98567    0.47  14           98.2       3     5
##  8       1376269     167 149426    4.02  14.5         93.9      14    39
##  9        717026     174  30000    3.39  15.9         96.1       3    98
## 10        102132     175 215000    0.26  16           88.4       6    95
## # … with 35 more rows, and 19 more variables: renovated <dbl>,
## #   class_a <dbl>, class_b <dbl>, LEED <dbl>, Energystar <dbl>,
## #   green_rating <dbl>, net <dbl>, amenities <dbl>, cd_total_07 <dbl>,
## #   hd_total07 <dbl>, total_dd_07 <dbl>, Precipitation <dbl>,
## #   Gas_Costs <dbl>, Electricity_Costs <dbl>, cluster_rent <dbl>,
## #   gas <dbl>, elec <dbl>, total_saved <dbl>, total_cost <dbl>
```

Here, we made some assumptions about the dataset:

cold days means demand for gas (heating), hot days means demand for electricity (cooling A/C)

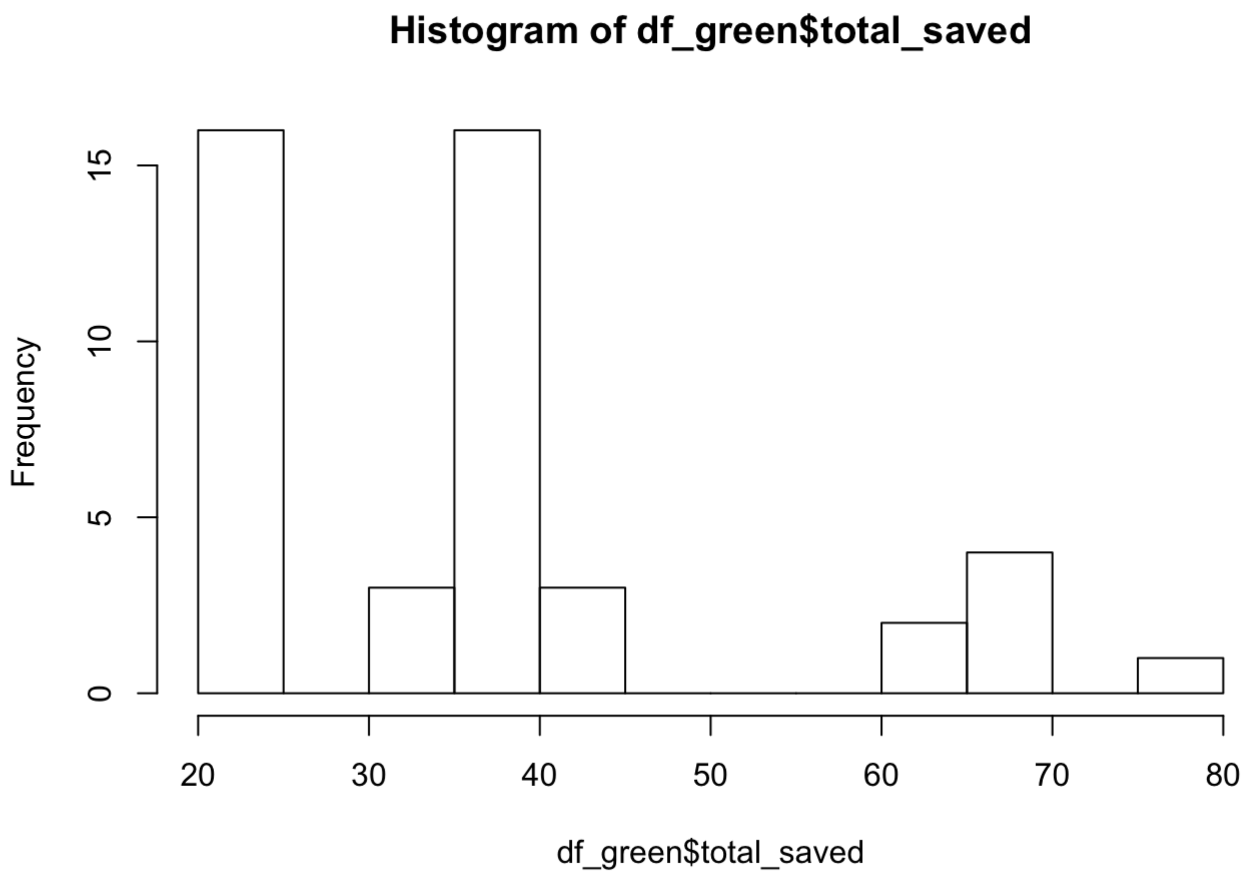
So our formulas are:

cd_total * Gas_Costs = gas bill per year hd_total * Electricity_Costs = electricity bill per year

These cost are per unit area per year.

We also assumed that green buildings in general can save 25% energy compare to non-green, according to numbers from LEED and National Geographics websites. So we multiple our cost by 0.25, and count it as the cost saved (extra revenue)

```
hist(df_green$total_saved, 10)
```




```
median_saved = median(df_green$total_saved)
yearly_saved = median_saved * 250000
yearly_saved
```

```
## [1] 9235600
```

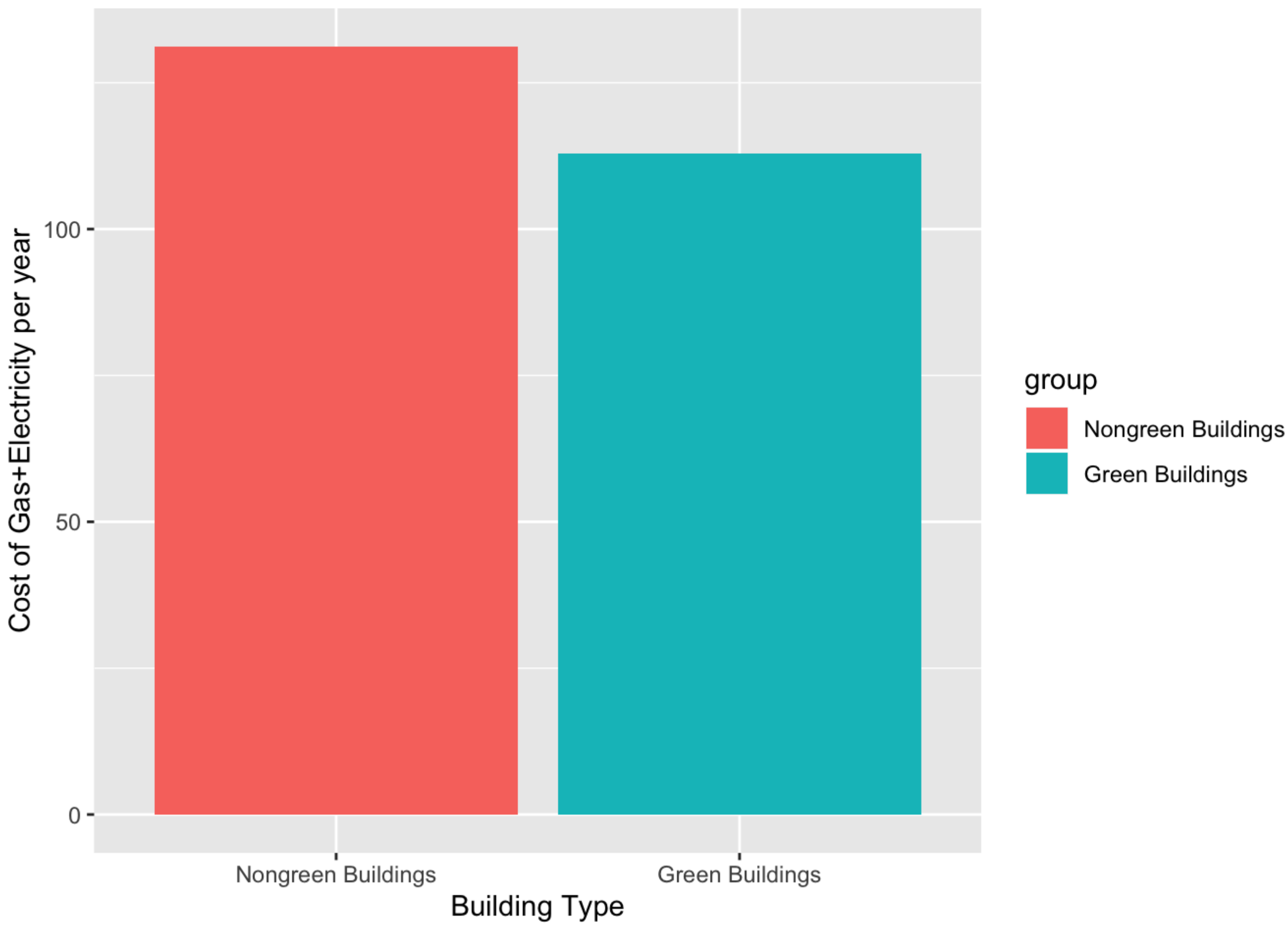
We calculated the median of total_saved in clusters that have similar weather in Austin, then multiplied it by our building's planned area in the problem, 250,000.

Our yearly saving on utility bills would be around 9,235,600.

```
green_mediancost = median(df_green$total_cost)
green_mean = mean(df_green$total_cost)

df_non_green$gas = df_non_green$cd_total_07 * df_non_green$Gas_Costs
df_non_green$elec = df_non_green$hd_total07 * df_non_green$Electricity_Costs
df_non_green$total_cost = (df_non_green$gas + df_non_green$elec)

nongreen_median_cost = median(df_non_green$total_cost)
nongreen_mean = mean(df_non_green$total_cost)

b = data.frame(group = 'Green Buildings', value = green_mean)
a = data.frame(group = 'Nongreen Buildings', value = nongreen_mean)
viz = rbind(a, b)
viz %>% group_by(group)
```

```
## # A tibble: 2 x 2
## # Groups:   group [2]
##   group               value
##   <fct>               <dbl>
## 1 Nongreen Buildings  131.
## 2 Green Buildings     113.
```

```
ggplot(viz, aes(x=group, y=value, fill=group)) +
  geom_col() + ylab('Cost of Gas+Electricity per year') + xlab("Building Type")
```

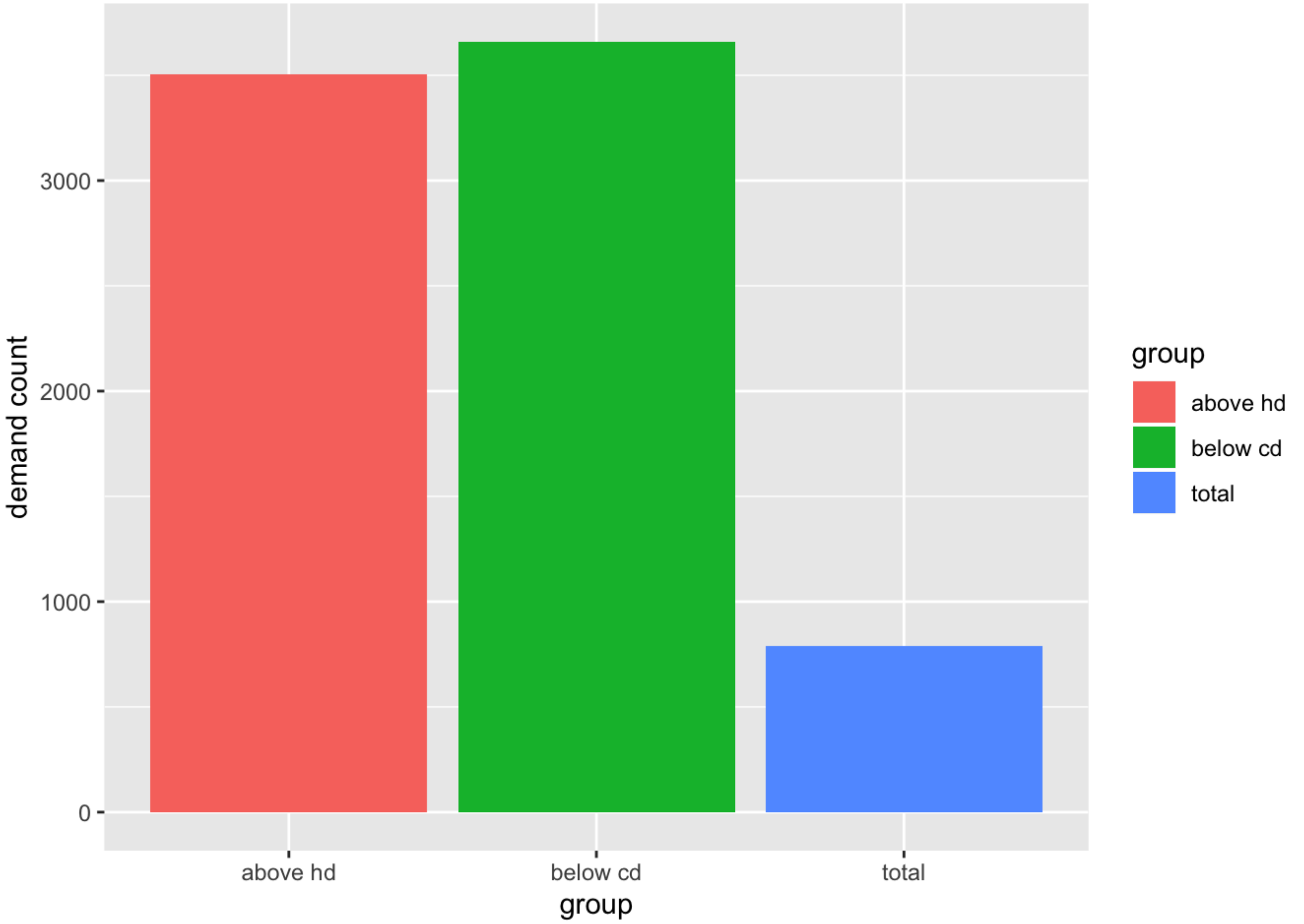



hd/cdboth counts bp

```
e = green_zero %>%
  filter(green_zero$hd_total07 > three_q)

r = green_zero  %>%
  filter(green_zero$cd_total_07 < one_q)

c= green_filt

g= data.frame(group= "above hd", value = nrow(e))
d = data.frame(group= "below cd", value = nrow(r))
c = data.frame(group = "total", value = nrow(c))

viz = rbind(g,d,c)
viz %>% group_by(group)
```

```
## # A tibble: 3 x 2
## # Groups:   group [3]
##   group    value
##   <fct>    <int>
## 1 above hd  3505
## 2 below cd  3660
## 3 total      788
```

```
ggplot(viz, aes(x=group, y=value, fill=group)) +
 geom_col() + ylab("demand count")
```




Because yearly extra saving(revenue) is around 9 million per year, the green building 5% premium fee 5 million extra cost will be recuperated in less than a year. On top of that, the building owner would save a lot more each year on the utility bills for the building running in general.

So based on our approach, we conclude that green building is a great idea. The immature data scientist's approach has the similar conclusion as ours, but he ignored too many factors.

Our approach explored deeper into the factor of weather, and utility cost.