R Notebook

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.1     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(readr)
library(viridis)
```

```
## Loading required package: viridisLite
```

```
##
## Attaching package: 'viridis'
```

```
## The following object is masked from 'package:viridisLite':
##
##     viridis.map
```

```
prediction_data <- read_csv("../data/prediction_data.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

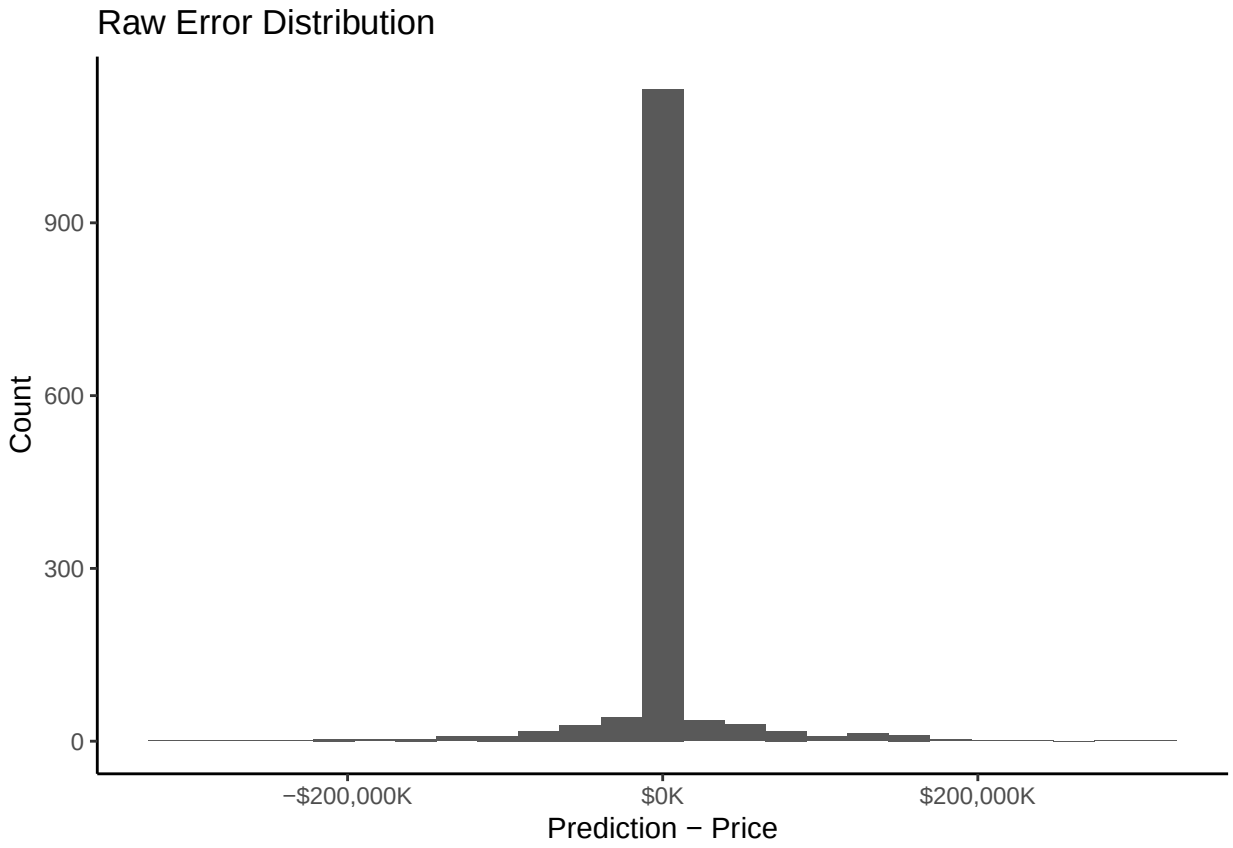
```
##
## -- Column specification --------------------------------------------------------
## cols(
##   X1 = col_double(),
##   bedrooms = col_double(),
##   price = col_double(),
##   bathrooms = col_double(),
##   sqft_living = col_double(),
##   sqft_lot = col_double(),
##   floors = col_double(),
##   waterfront = col_double(),
##   view = col_double(),
##   condition = col_double(),
##   sqft_above = col_double(),
##   sqft_basement = col_double(),
##   yr_built = col_double(),
##   yr_renovated = col_double(),
##   street = col_character(),
##   city = col_character(),
##   statezip = col_character(),
##   prediction = col_double()
```

```
## )
```

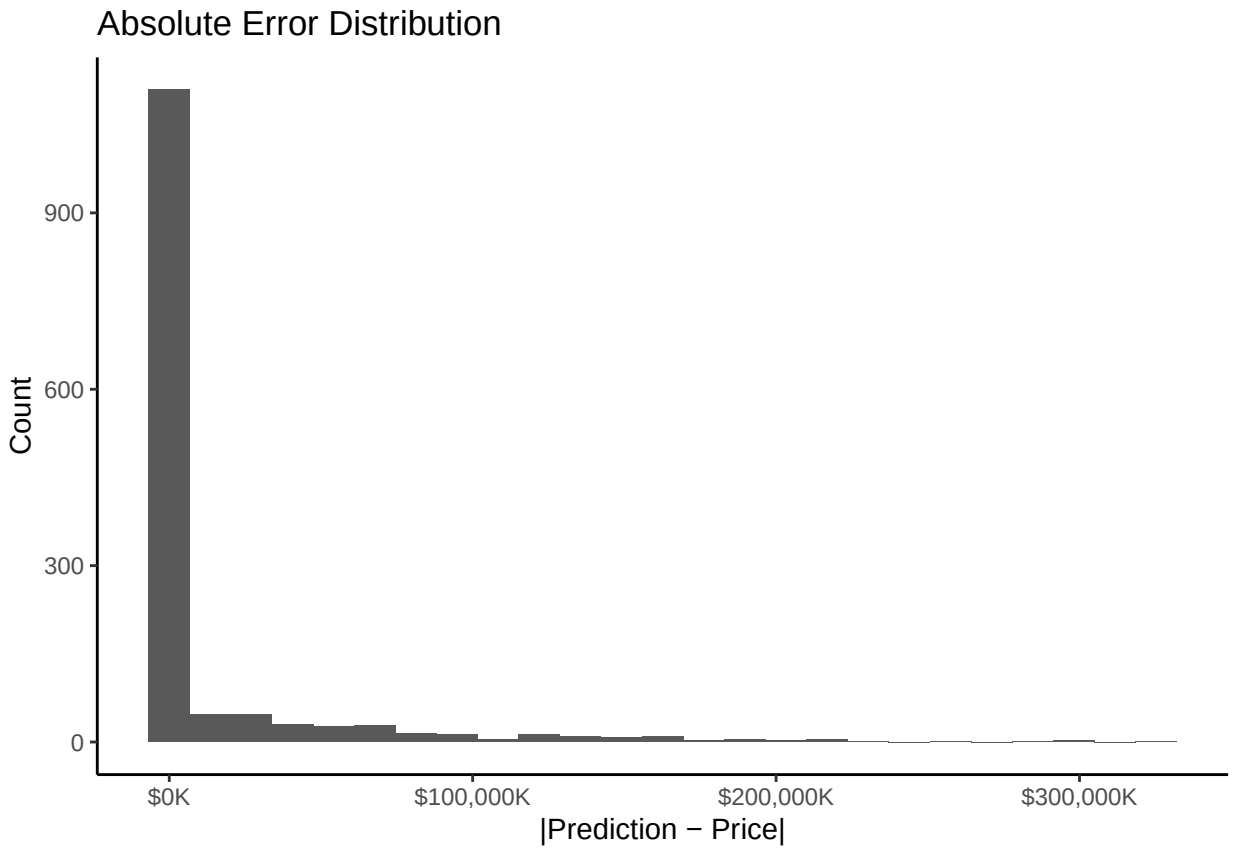
```
View(prediction_data)
```

```
prediction_data <- prediction_data %>%
  mutate(error = prediction - price,
         abs_error = abs(prediction - price),
         pct_error = abs_error  / price)
```

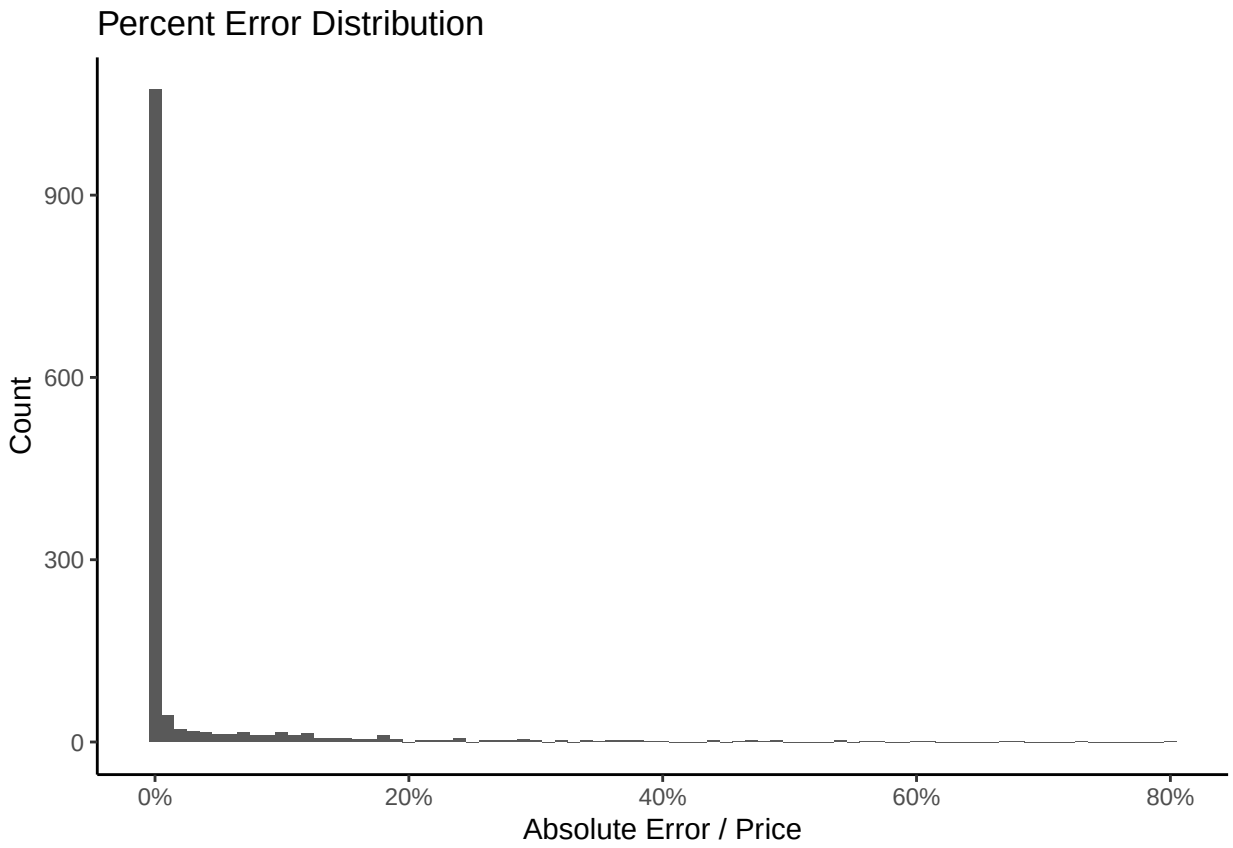
```
ggplot(prediction_data, aes(x=error)) +
  geom_histogram(bins=25) +
  scale_x_continuous(labels = scales::dollar_format(prefix="$", suffix = "K")) +
  xlab("Prediction - Price") +
  ylab("Count") +
  ggtitle("Raw Error Distribution") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme_classic()
```



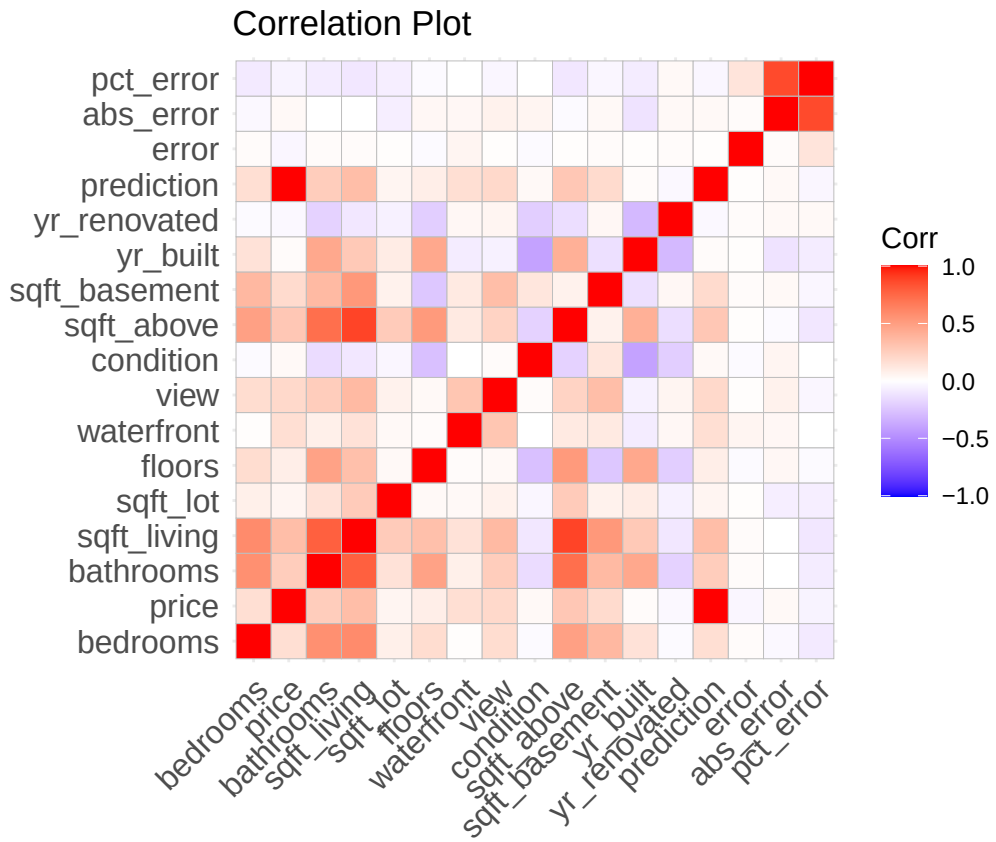

```
ggplot(prediction_data, aes(x=abs_error)) +
  geom_histogram(bins=25) +
  scale_x_continuous(labels = scales::dollar_format(prefix="$", suffix = "K")) +
  xlab("|Prediction - Price|") +
  ylab("Count") +
  ggtitle("Absolute Error Distribution") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme_classic()
```



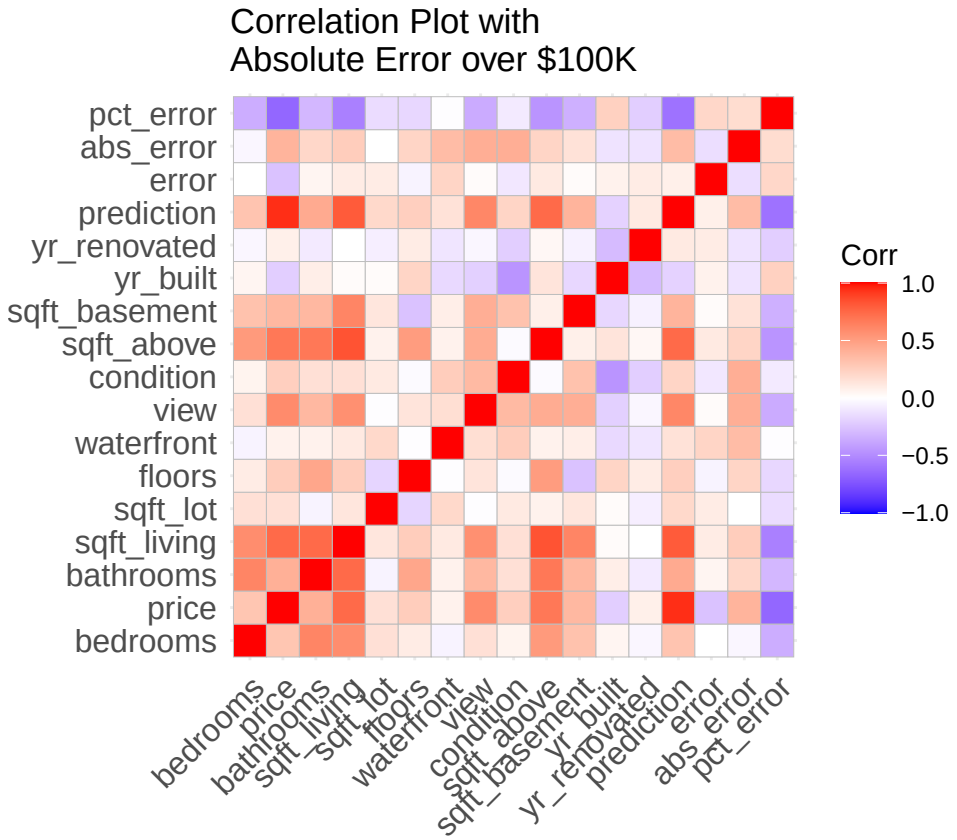

```
ggplot(prediction_data, aes(x=pct_error)) +
  geom_histogram(binwidth=.01) +
  scale_x_continuous(labels = scales::percent_format(suffix = "%")) +
  xlab("Absolute Error / Price") +
  ylab("Count") +
  ggtitle("Percent Error Distribution") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme_classic()
```




```
library(ggcorrplot)
corr_dat <- prediction_data %>%
  select(-X1, -street, -city, -statezip)
corr <- round(cor(corr_dat), 2)
ggcorrplot(corr, title = "Correlation Plot")
```




```
corr_dat <- prediction_data %>%
  select(-X1, -street, -city, -statezip) %>%
  filter(abs_error>100000)
corr <- round(cor(corr_dat), 2)
ggcorrplot(corr, title = "Correlation Plot with \nAbsolute Error over $100K")
```




```
summary(prediction_data$error)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -301100.7    -266.5       -2.6      435.8     208.6   325265.8
```

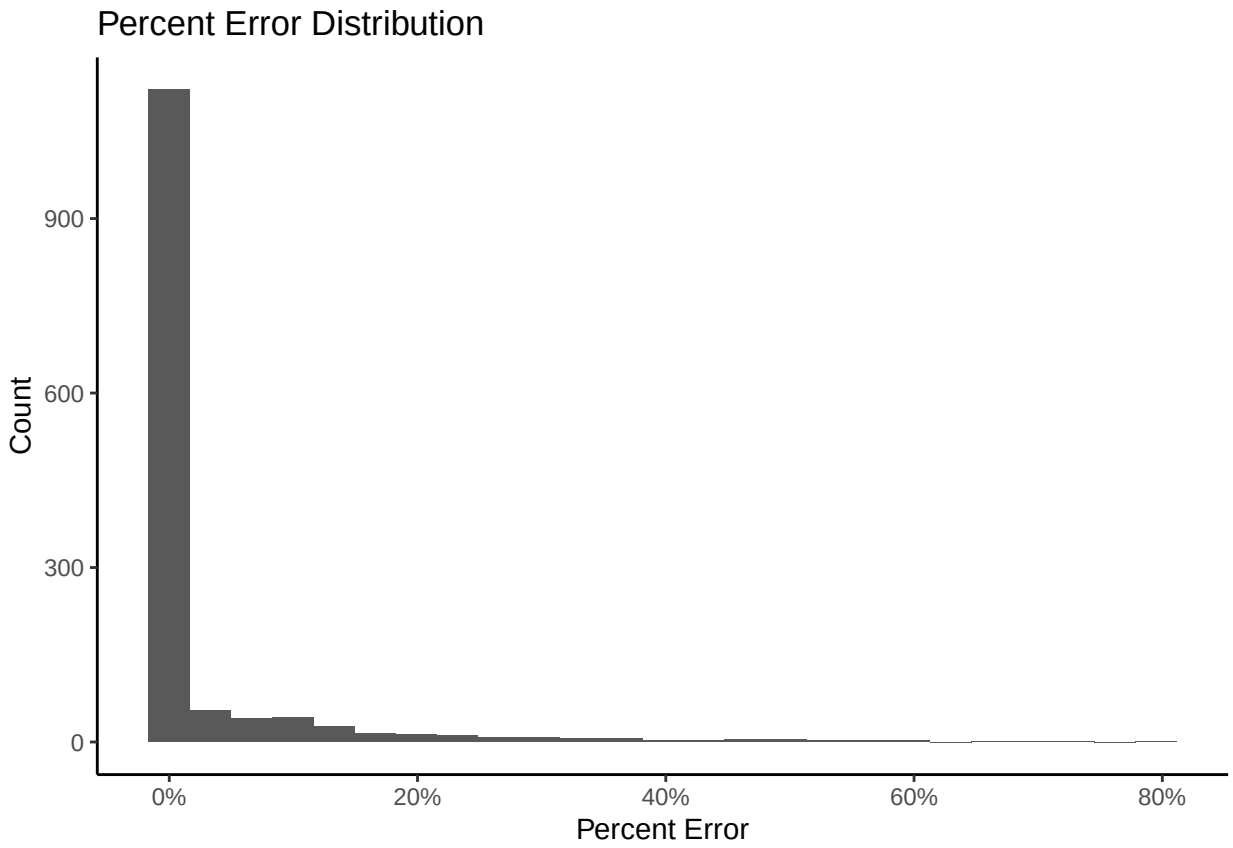
```
summary(prediction_data$abs_error)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##       0.1     96.9     234.4   13963.7    1182.5 325265.8
```

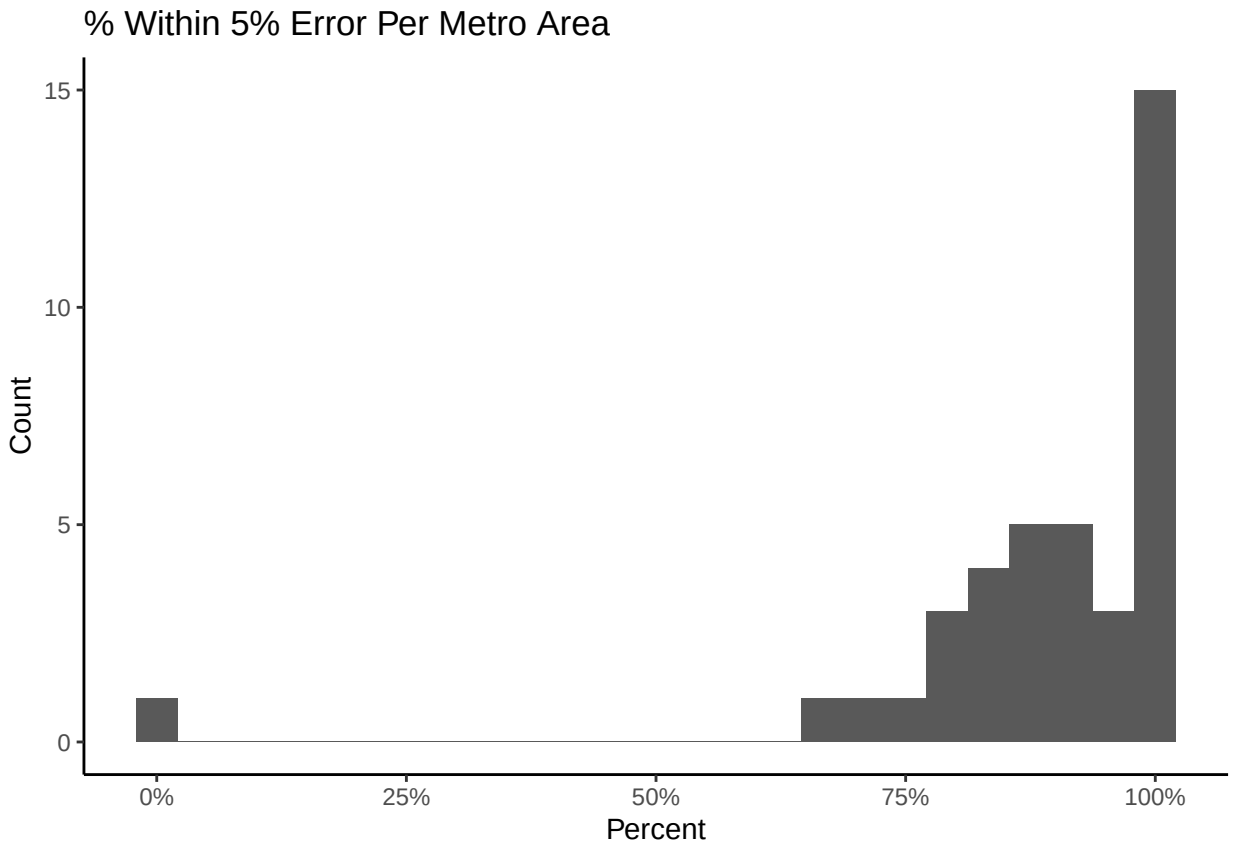
```
less <- nrow(prediction_data %>%
                 filter(pct_error<=.05))
less/nrow(prediction_data)
```

```
## [1] 0.8551674
```

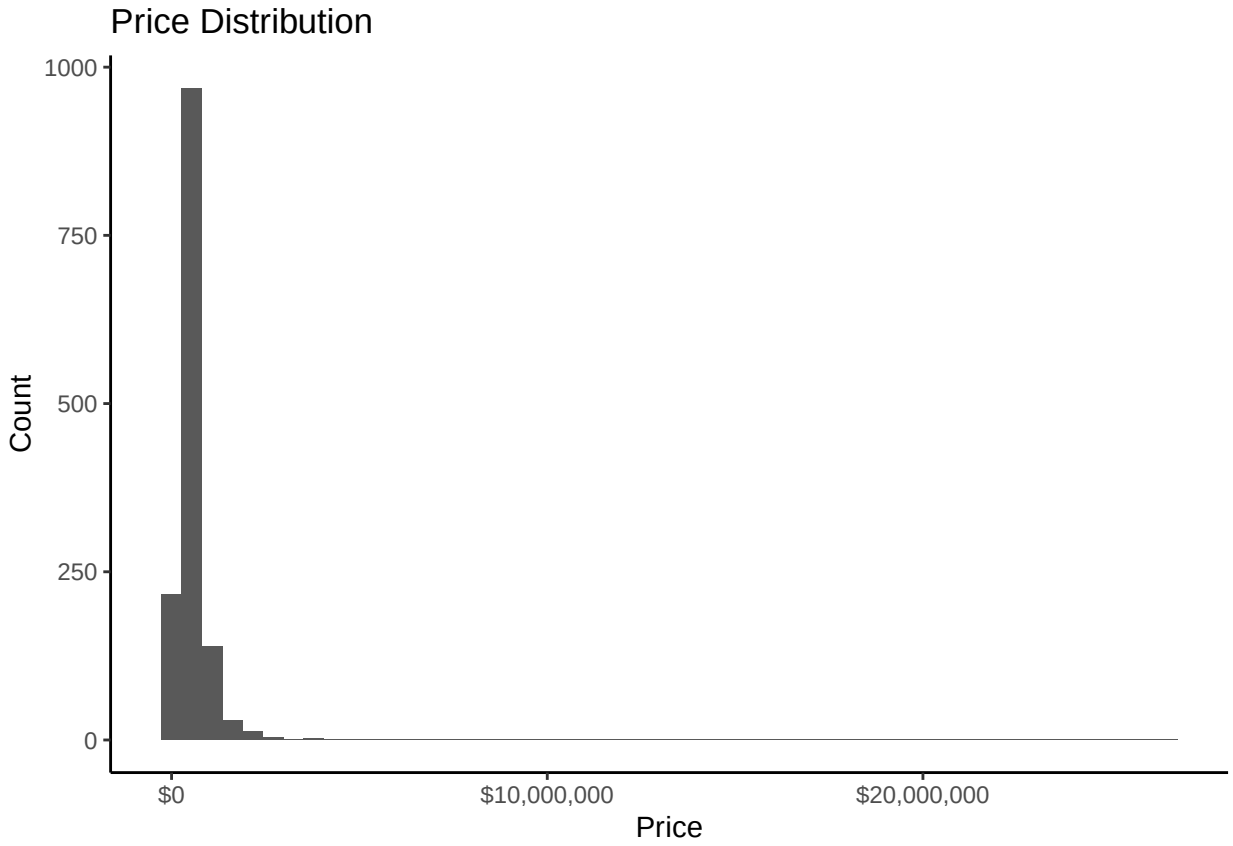
```
ggplot(prediction_data, aes(x=pct_error)) +
  geom_histogram(bins=25) +
  scale_x_continuous(labels = scales::percent_format(suffix = "%")) +
  xlab("Percent Error") +
  ylab("Count") +
  ggtitle("Percent Error Distribution") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme_classic()
```




```
prediction_data %>%
  group_by(city) %>%
  summarise(within5_pct = round(sum(pct_error<.05)/n(),2)) %>%
  ggplot(aes(x=within5_pct)) +
  geom_histogram(bins=25) +
  scale_x_continuous(labels = scales::percent_format(suffix = "%")) +
  xlab("Percent") +
  ylab("Count") +
  ggtitle("% Within 5% Error Per Metro Area") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme_classic()
```

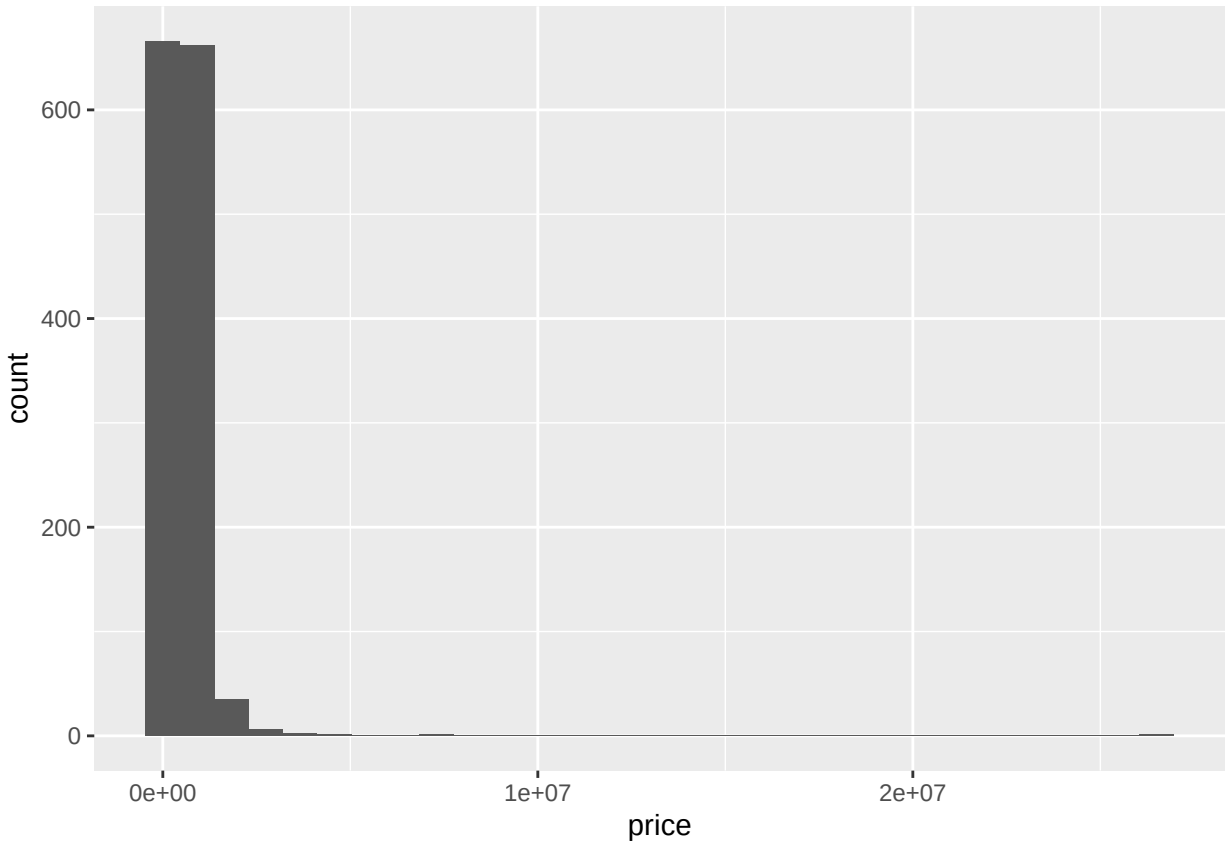



```
ggplot(prediction_data, aes(x=price)) +
  geom_histogram(bins=50) +
  scale_x_continuous(labels = scales::dollar_format(prefix="$")) +
  xlab("Price") +
  ylab("Count") +
  ggtitle("Price Distribution") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme_classic()
```




```
ggplot(prediction_data, aes(x=price)) +
  geom_histogram()
```

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


600
400
200
0
count
0e+00
1e+07
2e+07
price