BIRMINGHAM CITY
University

CMP5352 Report - TITLE NEEDED

DRAFT VERSION

Lewis Higgins - Student ID 22133848

Word count: XXXX

Abstract

text

text

text

Contents

Introduction

Text text text

Text text text

aaa

Motivation and objectives

```
library(ggplot2)

# Side-note that if you run this as a file and not in the IDE the plots will
# actually be put into a PDF for you in the active working directory (here).

# We can use a randomly selected sample of the dataset for the graph.
# However, because the results should be reproducible, we should set an RNG seed.
set.seed(1000)

# Select the numbered rows of the numbers produced by sample.
# Sample picks 100 random numbers.
dsmall <- diamonds[sample(nrow(diamonds), 100),]
# The random comma at the end tells R that you're slecting ROWS, not columns.
# If you don't put this comma it assumes you're looking to select the columns.

qplot(log(carat), log(price), data = dsmall, geom = "smooth")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

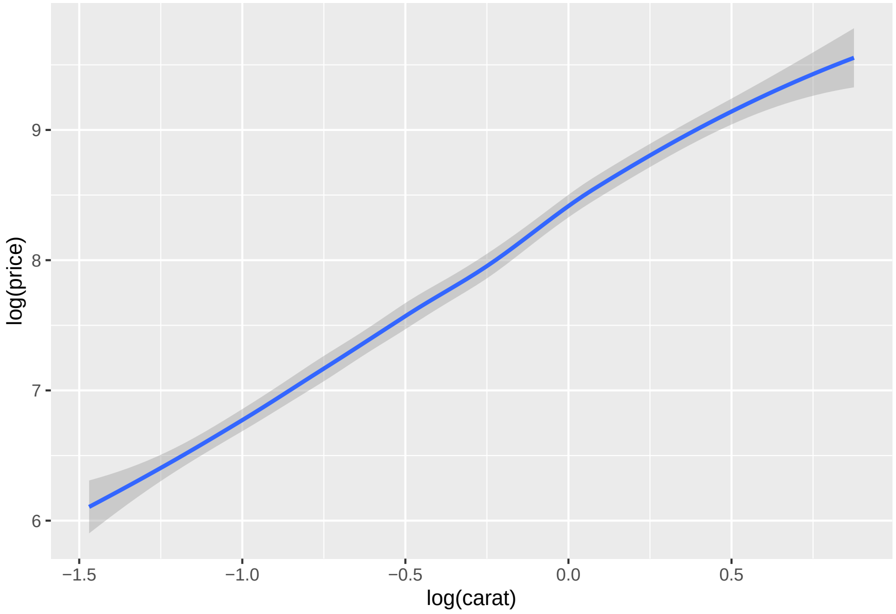

Figure 1: Figure AAA

```
# Can supply multiple geoms in a vector
qplot(log(carat), log(price), data = dsmall, geom = c("point", "smooth"))
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

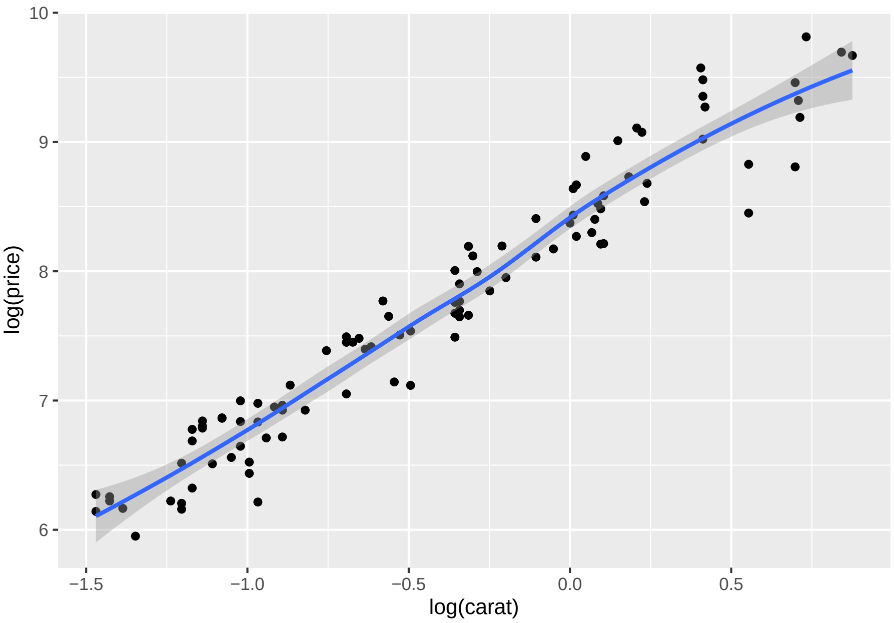

Figure 2: Figure AAA

```
# Different smooth line?

# Span varies the smoothness of geom_smooth from 0 to 1 where 1 is the smoothest.
# It states that span is an unknown parameter, yet this does actually
# modify the produced graph. 0.2 is the minimum before R throws warnings.
# 0.1 works with warnings, but anything lower produces no smooth line.
# Though, using 0.1 means you might as well not even put a smooth line.
qplot(log(carat), log(price), data = dsmall, geom = c("point", "smooth"), span = 0.2)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

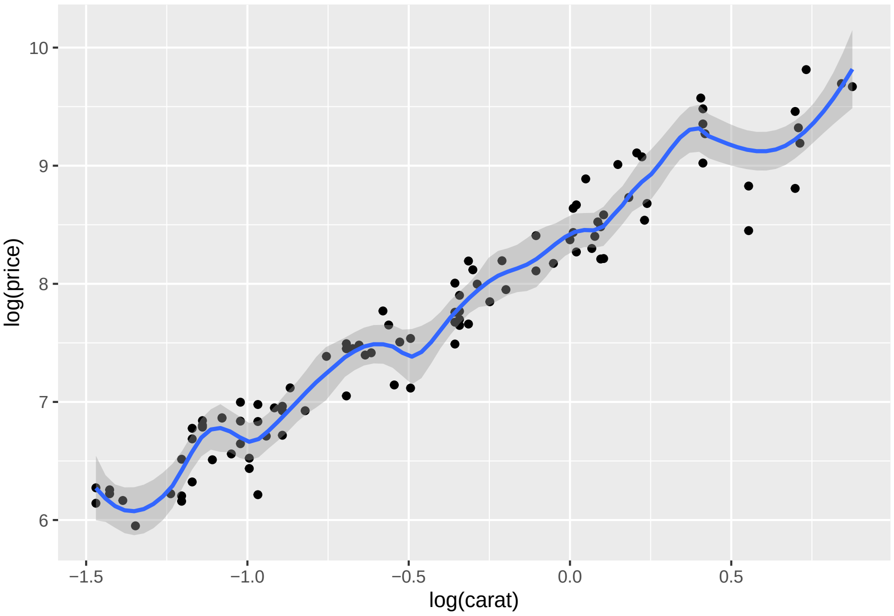

Figure 3: Figure AAA

```
# You can also fit a linear model to the graph via lm.
qplot(log(carat), log(price), data = dsmall, geom = c("point", "smooth"), method = "l
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

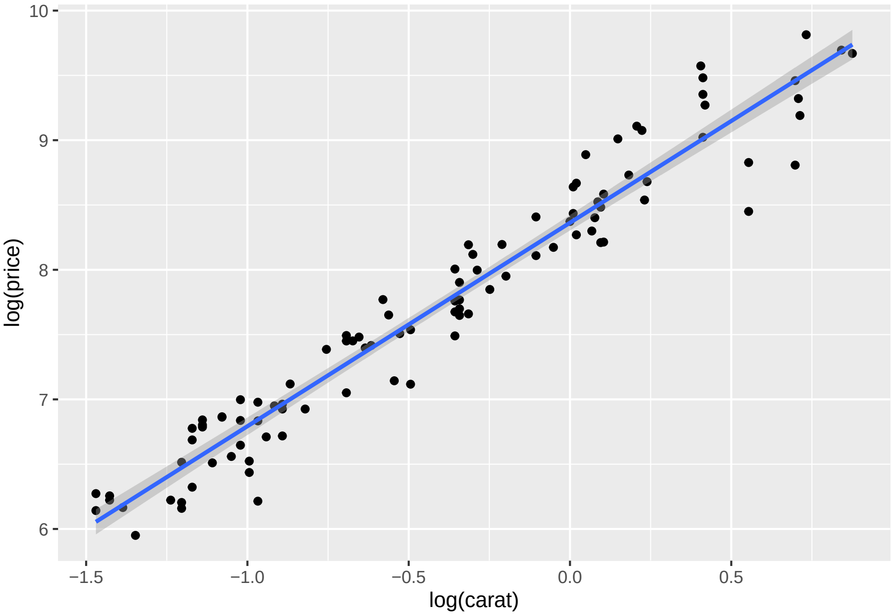

Figure 4: Figure AAA

```
# Scatterplotting a different dataset, ggplot's builtin mpg (car fuel economy data)
qplot(displ, hwy, data = mpg, color = drv)
```

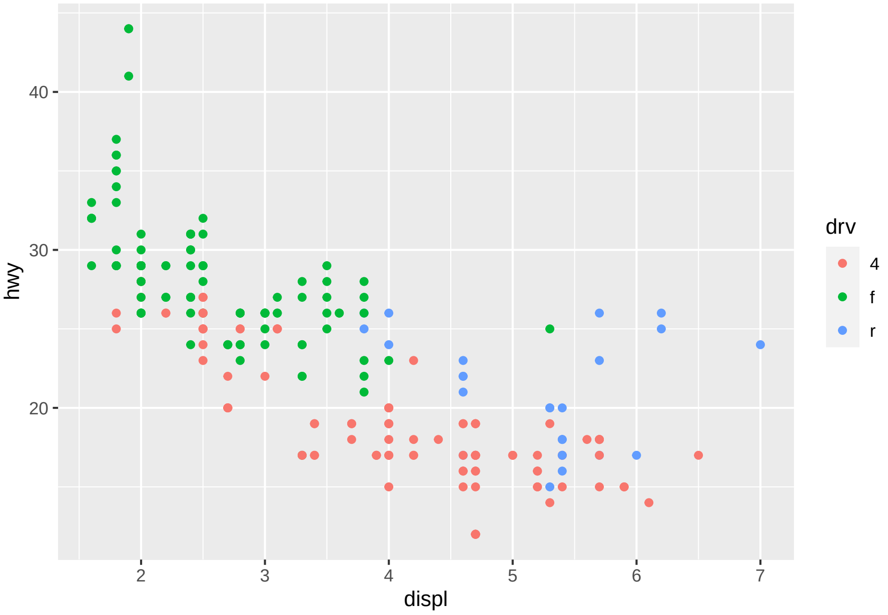

Figure 5: Figure AAA

```
# If you provided a color argument to this, it would draw one smooth for every color.
qplot(displ, hwy, data = mpg, geom = c("point", "smooth"))
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

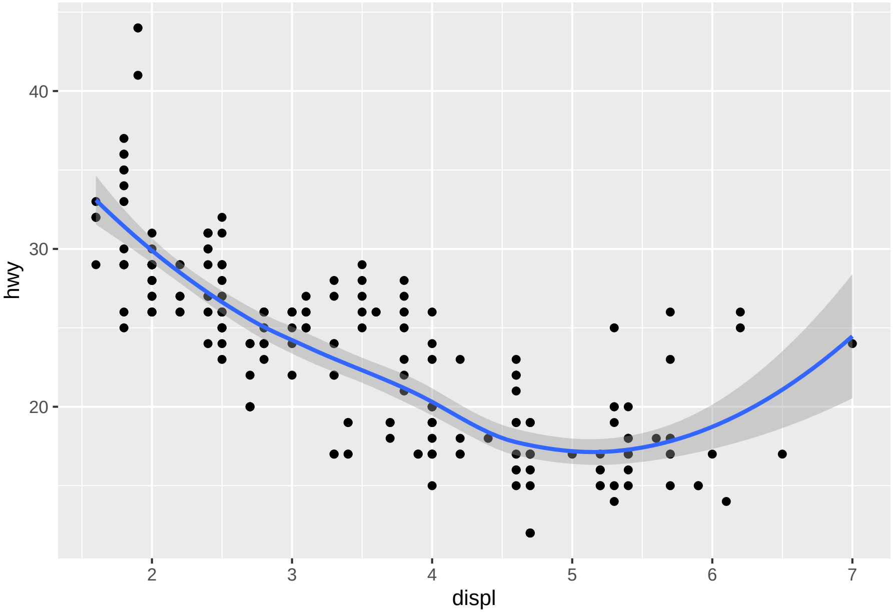

Figure 6: Figure AAA

```
# Answers the question "How are engine size and fuel economy related?"
# Turning cylinder into a factor (categorical data).
# Basically counts the appearances of each value.
qplot(displ, hwy, data = mpg, color = factor(cyl))
```

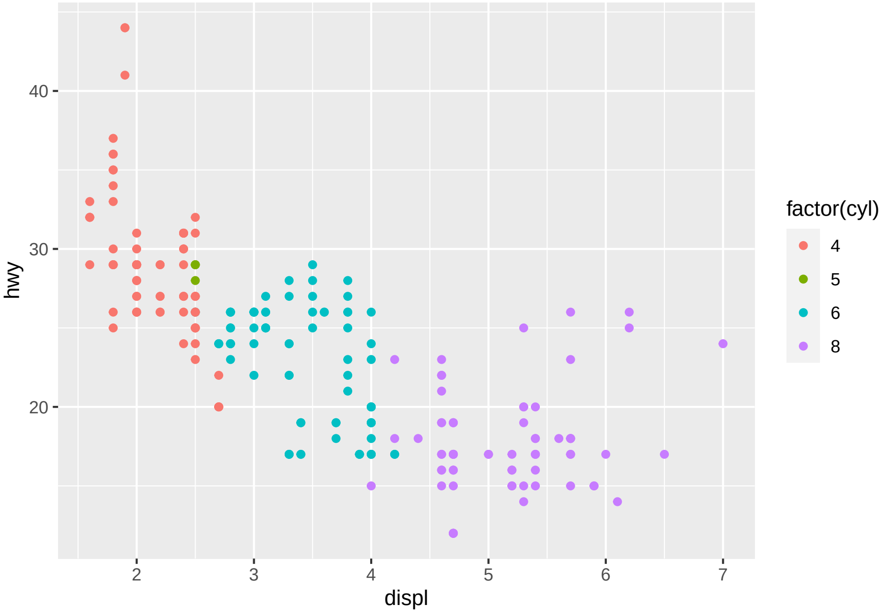

Figure 7: Figure AAA

```
# We can use all arguments previously shown at once.
# Note that there aren't enough 5 cylinders to fit a line, so there isn't one.
```

```
qplot(displ, hwy, data = mpg, color = factor(cyl),
      geom = c("point", "smooth"), method = "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

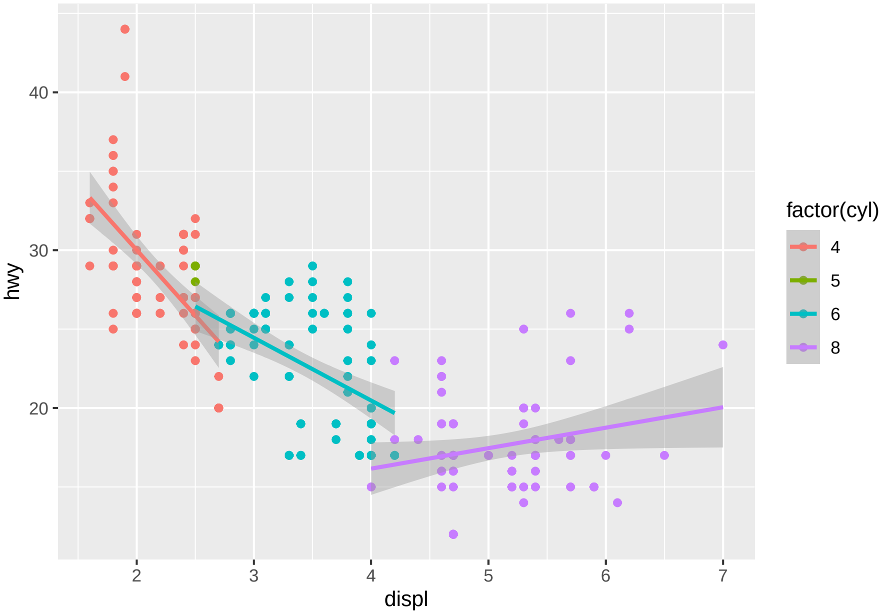

Figure 8: Figure AAA

```
### --- Faceting --- ###


# . acts as a placeholder, indicating that there's no variable.
# Results in three seperate histograms, one of each drive class.
qplot(hwy, data = mpg, facets = drv ~ ., binwidth = 2)
```

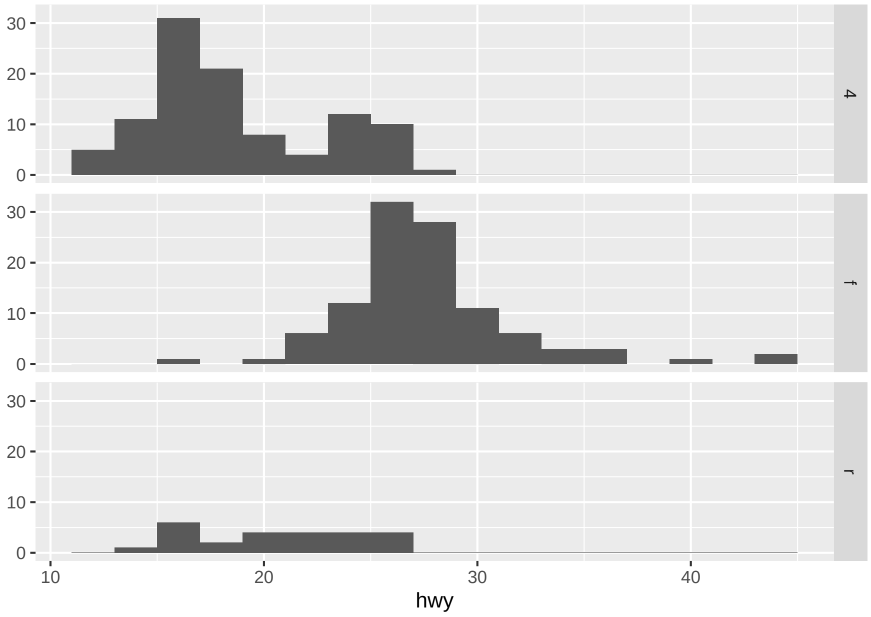

Figure 9: Figure AAA

```
# Could add colors. Doesn't help much though.

# Flips sideways. displ is displacement. Air movement per engine rev possibly
qplot(displ, hwy, data = mpg, facets = . ~ drv, geom = c("point", "smooth"))

## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

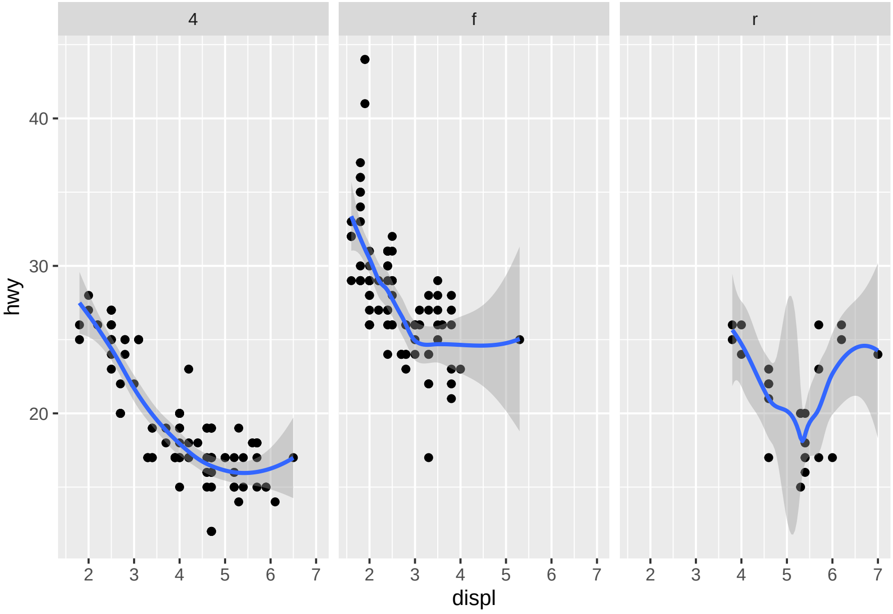

Figure 10: Figure AAA

```
# Reusing the diamond set.
qplot(carat, data = diamonds, facets = color ~ ., geom = "histogram")

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

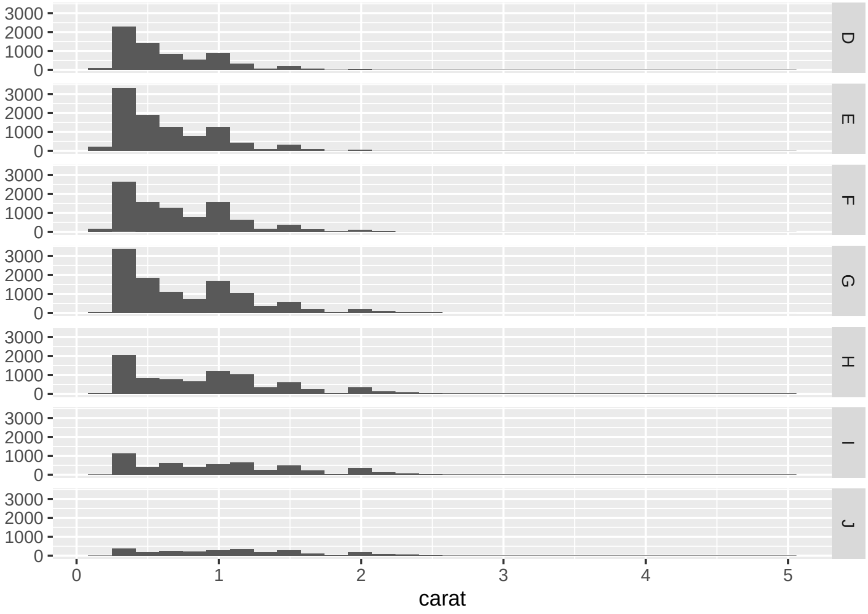

Figure 11: Figure AAA

```
# ..density.. tells ggplot to map the density as the Y-axis, instead of just counting
qplot(carat, ..density.., data = diamonds, facets = color ~ ., geom = "histogram")
```

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

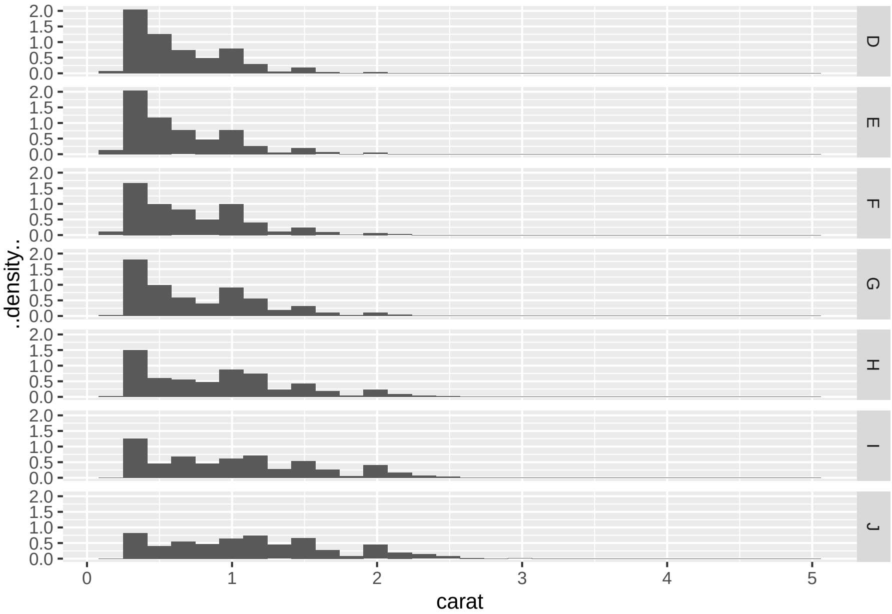

Figure 12: Figure AAA

```
# Plots thirty five histograms by also grouping by cut.
qplot(carat, data = diamonds, facets = color ~ cut, geom = "histogram")
```

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

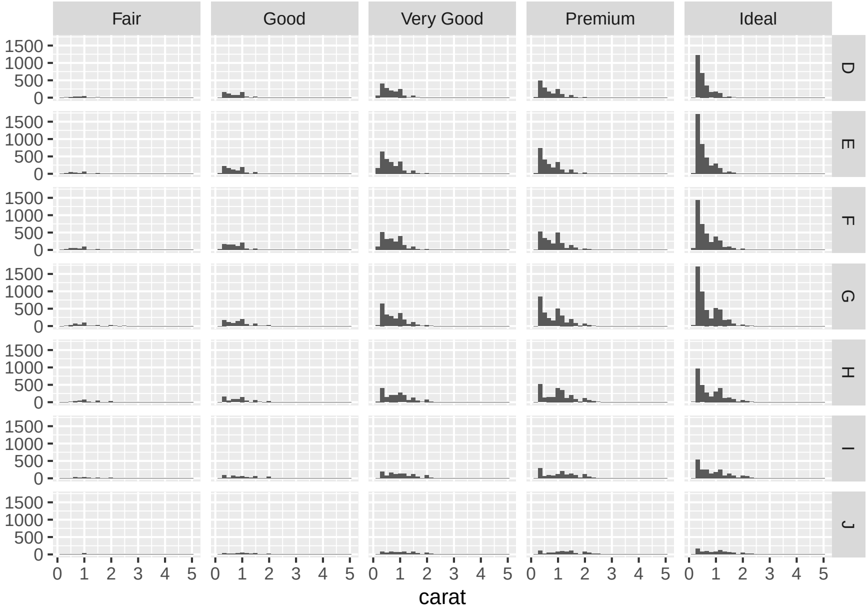

Figure 13: Figure AAA

Figure 14: A nice image.

Experimental results

text text text

text text text

Summary & conclusion

text text text

text text text