Petrol

Robert Schlegel & Ange Ackerman

22 April 2017

Introduction

This report shows analyses performed and figures created from 10+ years of petrol usage with a 2003 Volkswagen Polo sedan (1.4).

```
# Load libraries
library(tidyverse)
library(lubridate)
library(broom)

# Load data
petrol <- read.csv("/Users/morganbrand/Documents/WAS_CT_17/R_WAS/petrol_info.csv")
#petrol <- read.csv("~/R_WAS/petrol_info.csv")

# Correct date
petrol$date <- as.Date(petrol$date, "%d-%m-%y")

# Correct 'full' categorical label
petrol$full[is.na(petrol$full)] <- 0
petrol$full <- factor(petrol$full, labels = c("no", "yes"))
petrol$full <- factor(petrol$full, levels = c("yes", "no"))

# Remove problem rows
petrol <- petrol[complete.cases(petrol),]

# Add month and year columns
petrol$month <- floor_date(petrol$date, "month")
petrol$year <- floor_date(petrol$date, "year")
```

Odometre

Over the lifespan of a vehicle, one constant will always be the increasing count of the odometre. We may plot this as a time series.

```
ggplot(data = petrol, aes(x = date, y = odom)) +
  geom_line() +
  geom_point() +
  scale_y_continuous(breaks = seq(80000, 200000, 20000)) +
  labs(y = "distance (km)", x = "")
```

Also of interest is how far the distances between fill ups may be. There are a couple of gaps in this time series, which lend themselves to some dramatic numbers, but overall we are able to get an idea of the true distances.

```
# Create a column showing distance between fill-ups
petrol$dist <- c(0,diff(as.matrix(petrol$odom)))
```

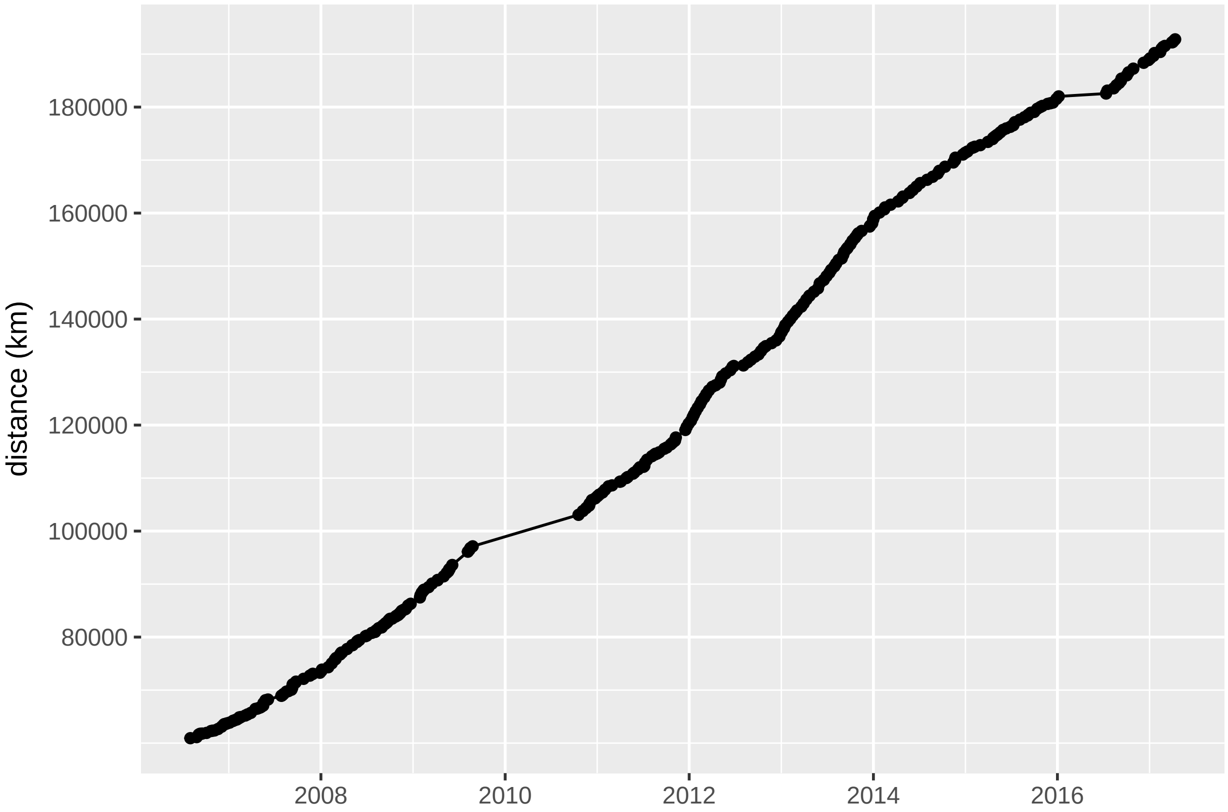


Figure 1: Line graph showing total distance traveled (km) over time.

```r
# Total distance traveled
petrol$dist_total <- petrol$odom-petrol$odom[1]

# Histogram
ggplot(data = petrol, aes(x = date, y = odom)) +
  geom_line() +
  geom_point() +
  geom_line(aes(y = dist), colour = "blue") +
  scale_y_continuous(breaks = seq(0, 200000, 40000)) +
  labs(y = "distance (km)", x = "")
```

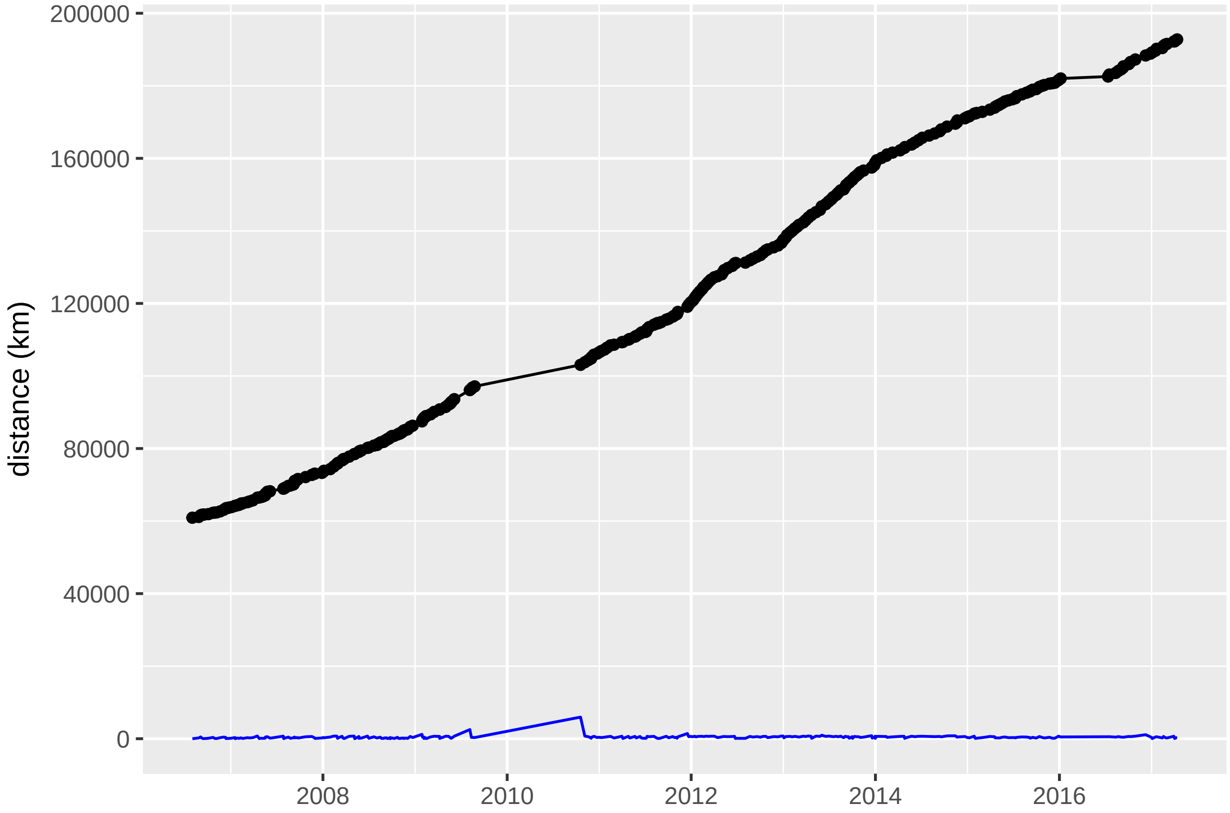


Figure 2: Same as previous figure with the total distance between fill ups shown with a blue line.

This blue line would provide more useful information if visualised as a histogram.

```r
ggplot(data = petrol[petrol$dist <= 1000,], aes(x = dist)) +
  geom_histogram(fill = "violet", colour = "grey40")
```

This histogram shows a bimodal distribution with a clustering of distances around 100 kms and 600 kms. This is not so strange if you think about it as it shows that this driver would tend to either go short distance between filling up, or long distances. This is likely linked to spending behaviours, which is the next thing to investigate.

Spending patterns

We may produce another histogram to plot the amount of money spent per visit to the petrol station.

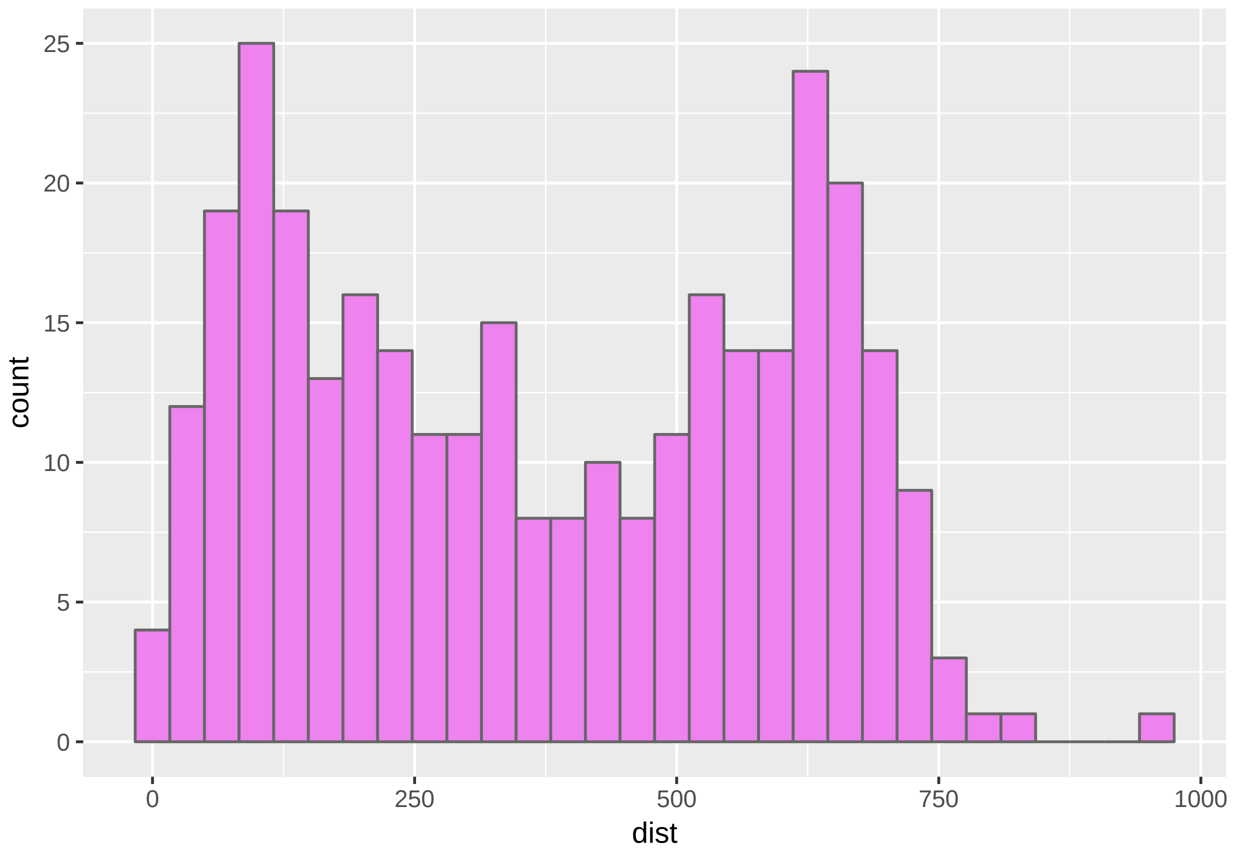


Figure 3: Histogram showing the distances traveled between fillings. Any values over 1,000km were subsetted out.

```
ggplot(data = petrol, aes(x = cost)) +
  geom_histogram(aes(fill = full), colour = "grey40") +
  labs(x = "Price (R)")
```

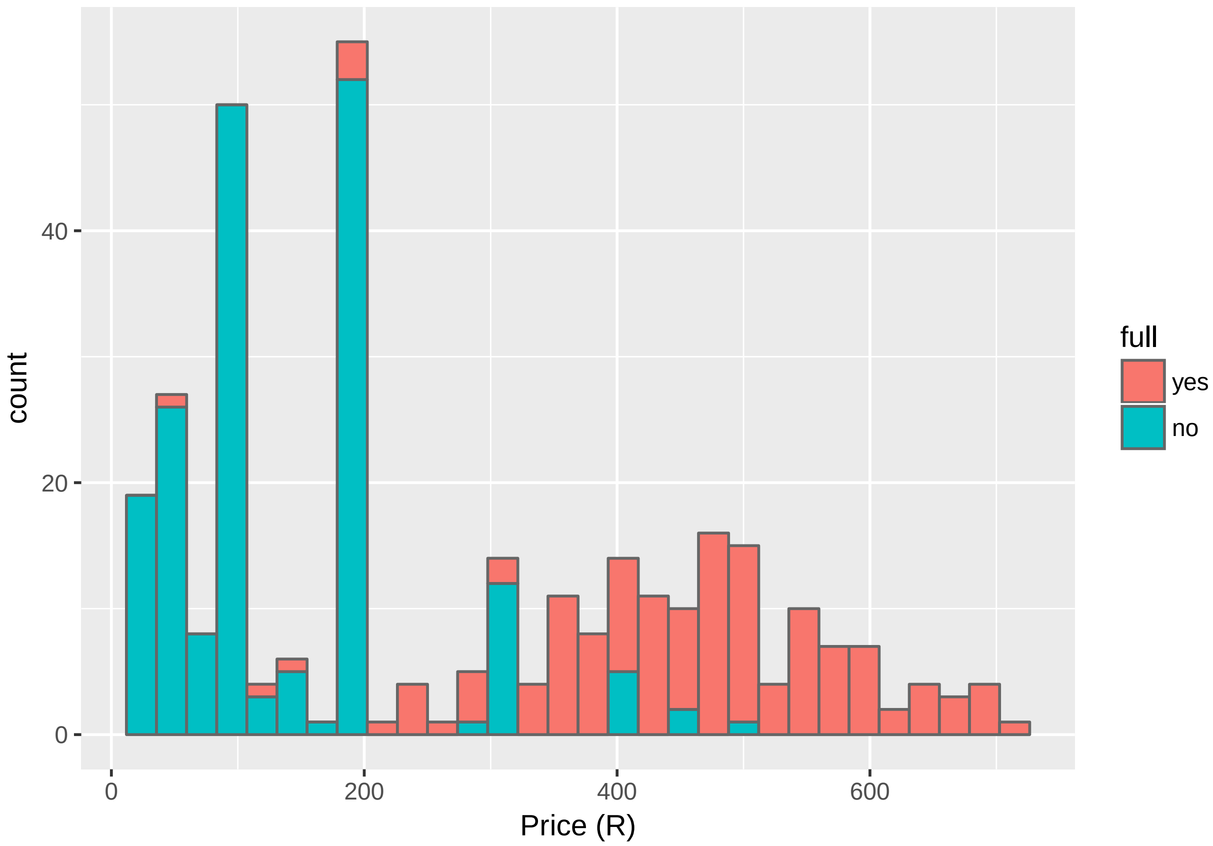


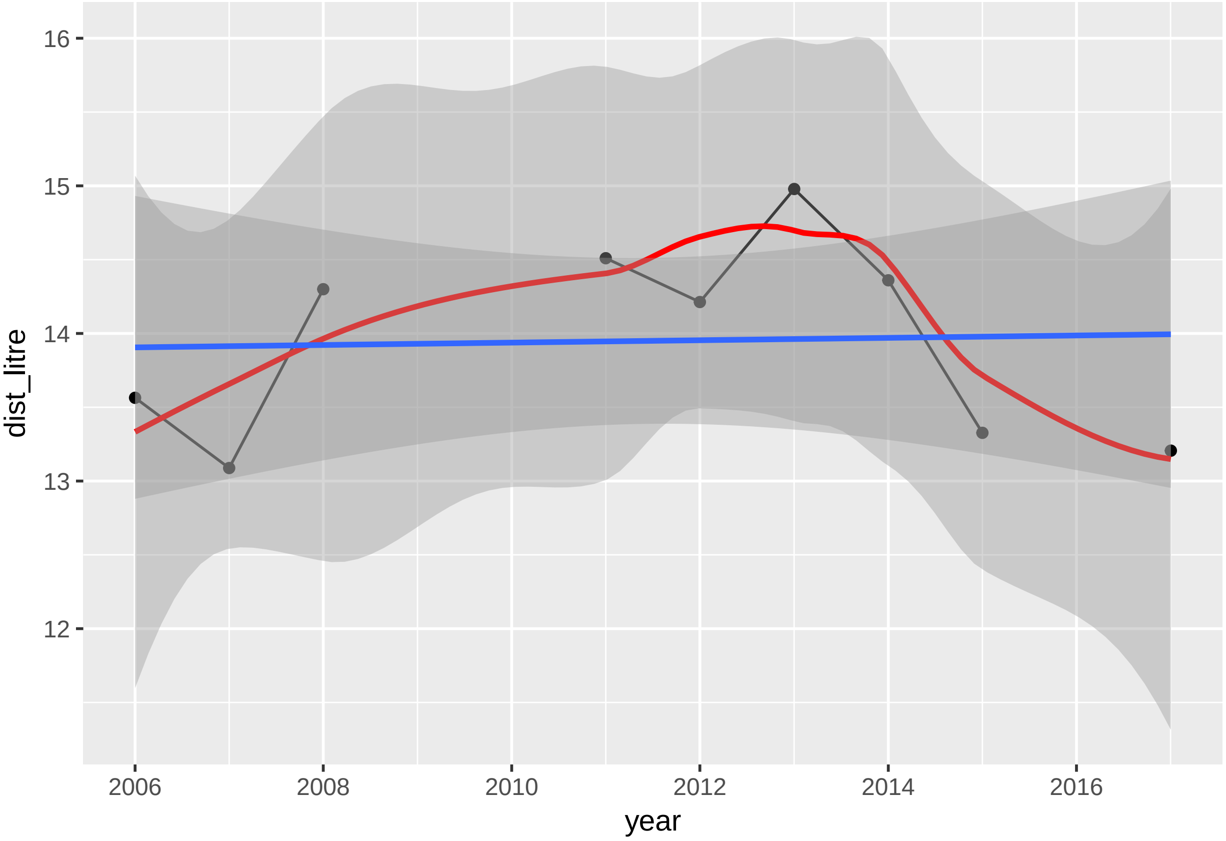
Figure 4: Histogram showing the amount of money spent per visit to the petrol station. The colours show if the tank was filled during that visit or not.

The distribution shown in this histogram is also not surprising if one thinks about it. This histogram is showing two different spending habits. On the left hand side we see that there are distinct columns rising out from the others. This is when the driver went to the station and spent specifically, R20, R50, R100, R200, R300 or R400. On the right hand side of the histogram we see a more normal distribution of columns. These are the prices spent when filling up the tank to full.

Petrol usage per km

One of the first things any car owner wants to know about their vehicle is the mileage their vehicle is getting. And whether or not this is decreasing with wear. Because we don't know exactly how much petrol is used between each fill up this becomes a bit tricky. We overcome this challenge by creating annual sums of petrol use. With these we may then calculate the distance traveled per litre more broadly. Monthly means are too erratic to be useful.

```
# Create monthly means
petrol_annual <- petrol %>%
  select(-full, -date, -month) %>%
  group_by(year) %>%
  mutate(dist2 = sum(dist)) %>%
  mutate(litre2 = sum (litre)) %>%
  summarise_all(mean) %>%
  mutate(dist_litre = dist2/litre2)

# Remove outliers caused during absences
is.na(petrol_annual$dist_litre) <- petrol_annual$dist_litre > 16

# Plot it
ggplot(data = petrol_annual, aes (x = year, y = dist_litre)) +
  geom_line() +
  geom_point() +
  geom_smooth(colour = "red") +
  geom_smooth(method = "lm")
```



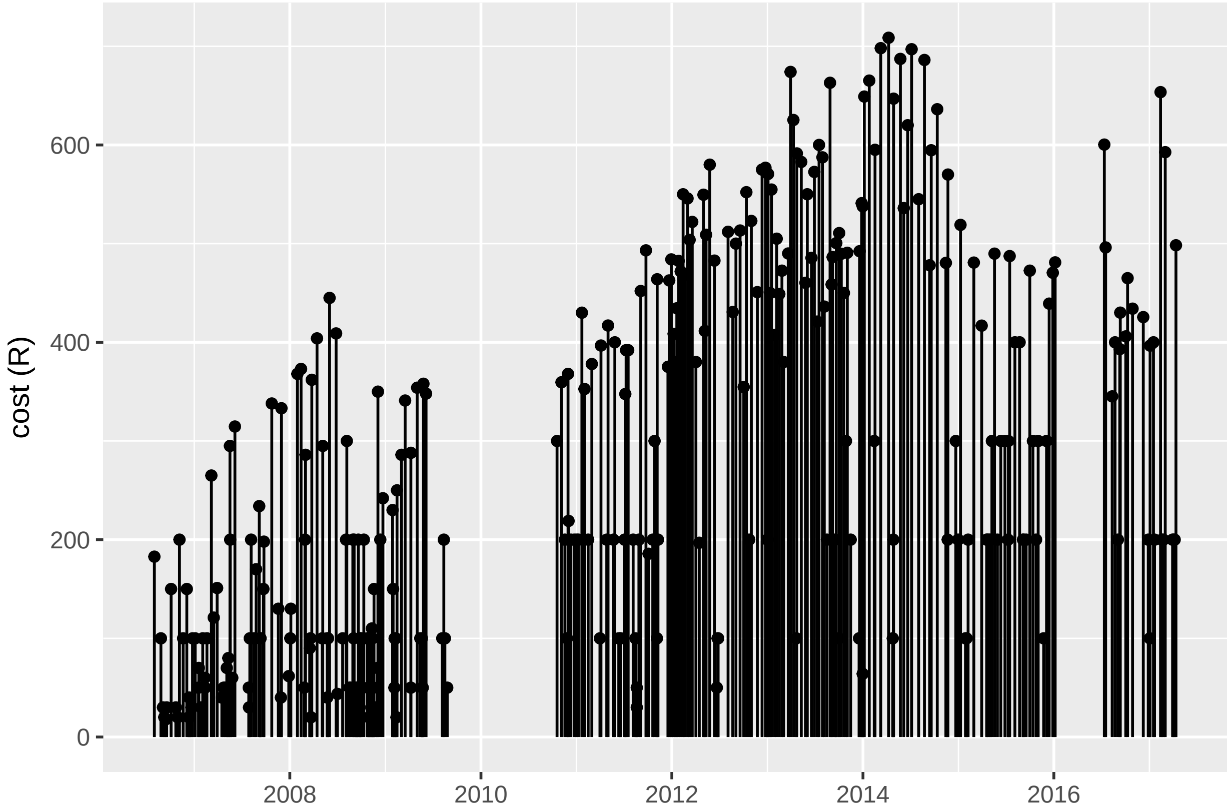
As we may see, the mileage appears to increase until 2013 when it then falls precipitously. The overall change in mileage for this car appears flat when modeled linearly.

Petrol prices

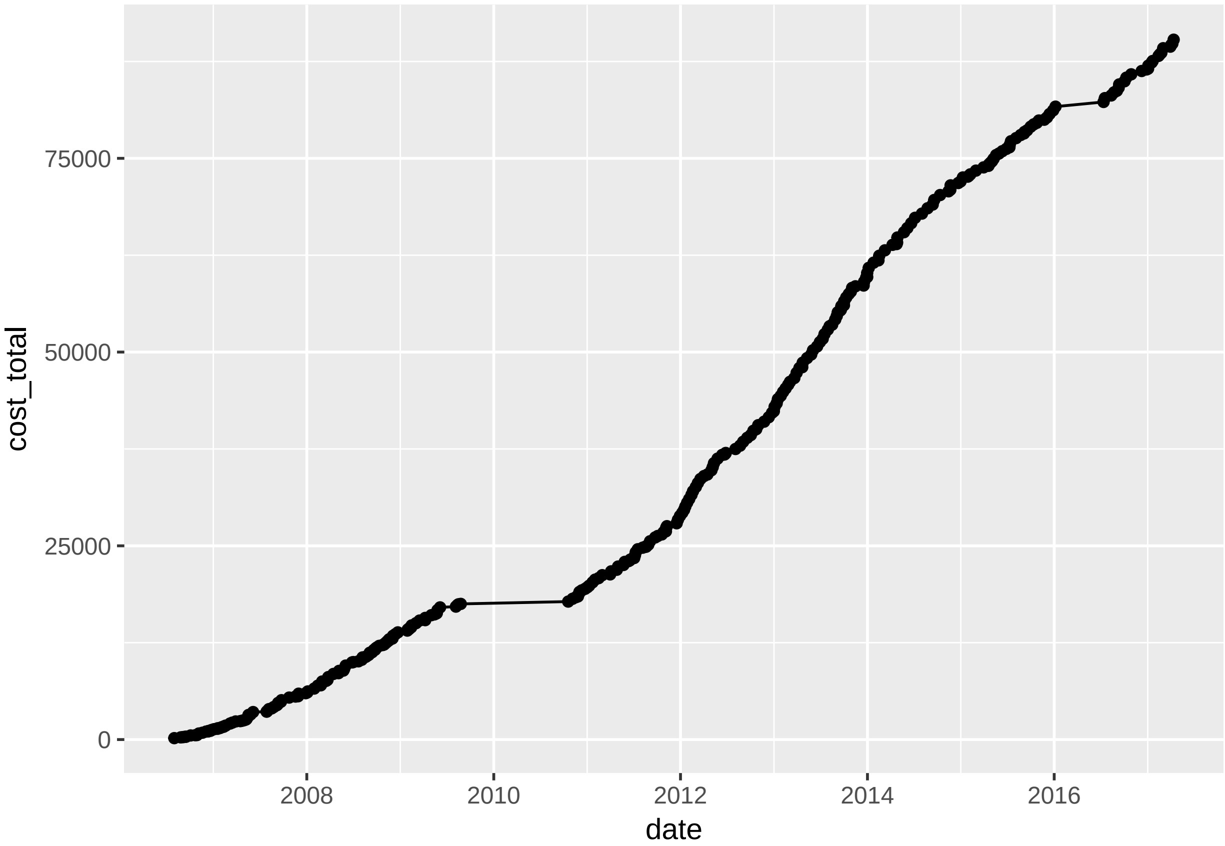
Of interest to everyone is the price of petrol. Both how much we have spent and how much we have to spend every time we pull up to the station. First we see a lolliplot of spending behaviour.

```
# Calculate total amount spent
petrol$cost_total <- cumsum(petrol$cost)
```

```
# Lolli plot
ggplot(data = petrol, aes(x = date, y = cost)) +
  geom_point() +
  geom_segment(aes(xend = date, y = 0, yend = cost)) +
  labs(y = "cost (R)", x = "")
```



```
ggplot(data = petrol, aes(x = date, y = cost_total)) +
  geom_line() +
  geom_point()
```

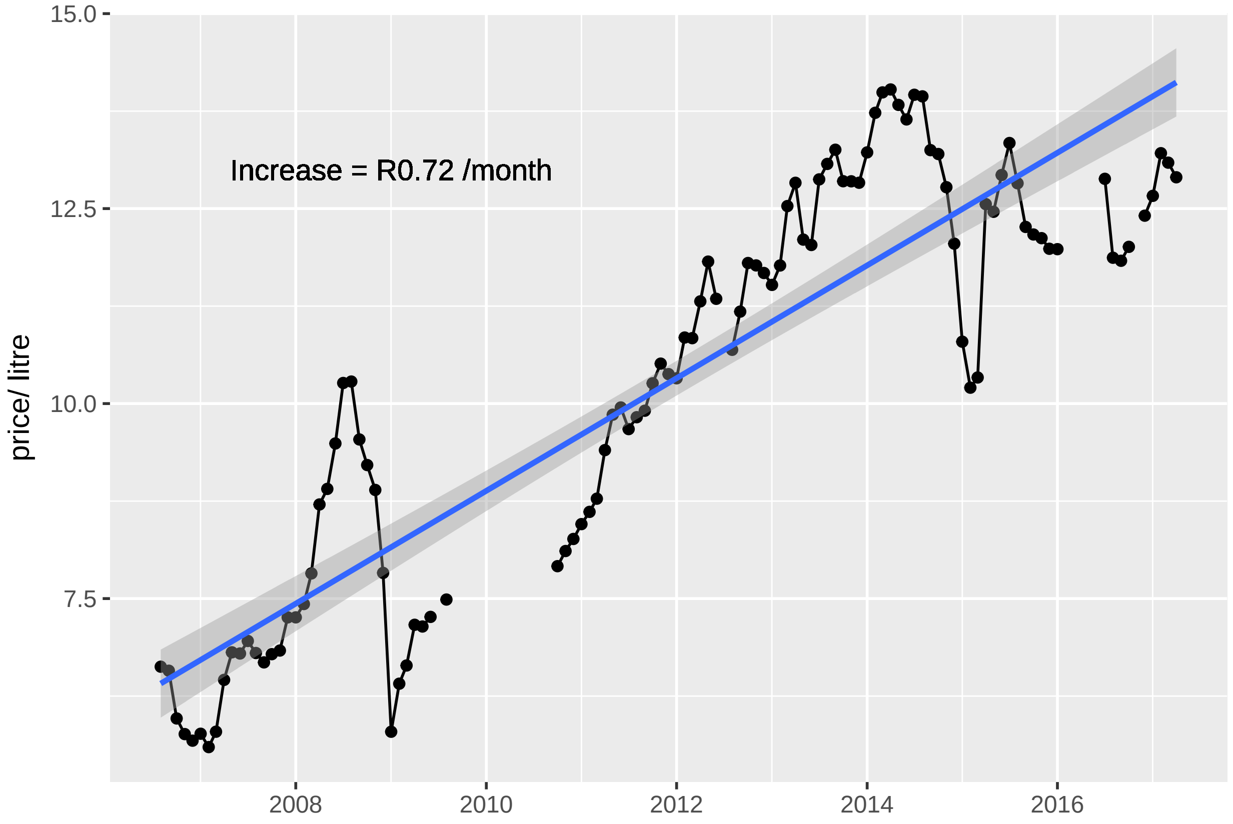


And then the price per litre averaged per month.

```r
# Price/ litre/ month
petrol_monthly <- petrol %>%
  select(-full, -date, -year) %>%
  group_by(month) %>%
  summarise_all(mean) %>%
  mutate(price_litre = cost/litre)

# Fill in missing months
month_index <- data.frame(month = seq(petrol_monthly$month[1], petrol_monthly$month[nrow(petrol_monthly)
petrol_monthly <- merge(petrol_monthly, month_index, by = "month", all.y = TRUE)

petrol_trend <- lm(petrol_monthly$price_litre ~ seq(1:nrow(petrol_monthly)))
petrol_augment <- augment(petrol_trend)
petrol_tidy <- tidy(petrol_trend)
petrol_glance <- glance(petrol_trend)
petrol_tidy$estimate[2]*12
```

```
## [1] 0.7226146
```

```r
# R 0.72/ month

# Line graph
ggplot(data = petrol_monthly, aes(x = month, y = price_litre)) +
  geom_line() +
  geom_point() +
  geom_smooth(method = "lm") +
  geom_text(aes(x = as.Date("2009-01-01"), y = 13,
                label = paste0("Increase = R", round(petrol_tidy$estimate[2]*12, 2), " /month"))) +
  # geom_point(data = petrol[187,], colour = "red")
  # scale_y_continuous(breaks = seq(80000, 200000, 20000)) +
  labs(y = "price/ litre", x = "")
```


Increase = R0.72 /month
price/ litre