```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────────────────────
──────────────────── tidyverse 1.3.0 ──
```

```
## ✓ ggplot2 3.3.2     ✓ purrr   0.3.4
## ✓ tibble  3.0.3     ✓ dplyr   1.0.2
## ✓ tidyr   1.1.2     ✓ stringr 1.4.0
## ✓ readr   1.3.1     ✓ forcats 0.5.0
```

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

```
library(ggdark)
```

Load in the data

```
wine_data <- read_csv("./data/BordeauxWines.csv", locale = readr::locale(encoding = "lat
in1"))
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   Wine = col_character(),
##   Price = col_character()
## )
```

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

```
# summary(wine_data)
# str(wine_data)
```

Mutate our variables into factors

```
wine_cols <- c(5:989)
wine_data[,wine_cols] <- lapply(wine_data[,wine_cols], factor)
#wine_data[,wine_cols] <- lapply(wine_data[,wine_cols], factor, level = c(0, 1))

# ISSUE: when factorized, wine sometimes has columns with only 1 factor, this selects on
ly columns with multiple factors and drops the rest
wine_fixed <- wine_data[, sapply(wine_data, function(col) length(unique(col))) > 1]

# str(wine_fixed, list.len=ncol(wine_fixed))
```

Splice our dataset to make it more manageable when cleaning.

```
wine_spliced <- wine_fixed[1:10,]
```

Remove dollar signs from Price.

```
wine_fixed$Price <- str_replace(wine_fixed$Price, "\\$", "")

wine_fixed$Price <- as.numeric(wine_fixed$Price)
```

```
## Warning: NAs introduced by coercion
```

```
# wine_spliced$Price <- gsub("\\$", "", wine_spliced$Price)
```

Visualizations

```
price_score_plot <- ggplot(wine_fixed, aes(x = Price, y = Score)) +
  geom_point() +
  geom_smooth(method = "lm", color = "red") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), # Turn of the background grid
    panel.grid.minor = element_blank(),
    panel.background = element_blank()) +
  dark_theme_gray()
```

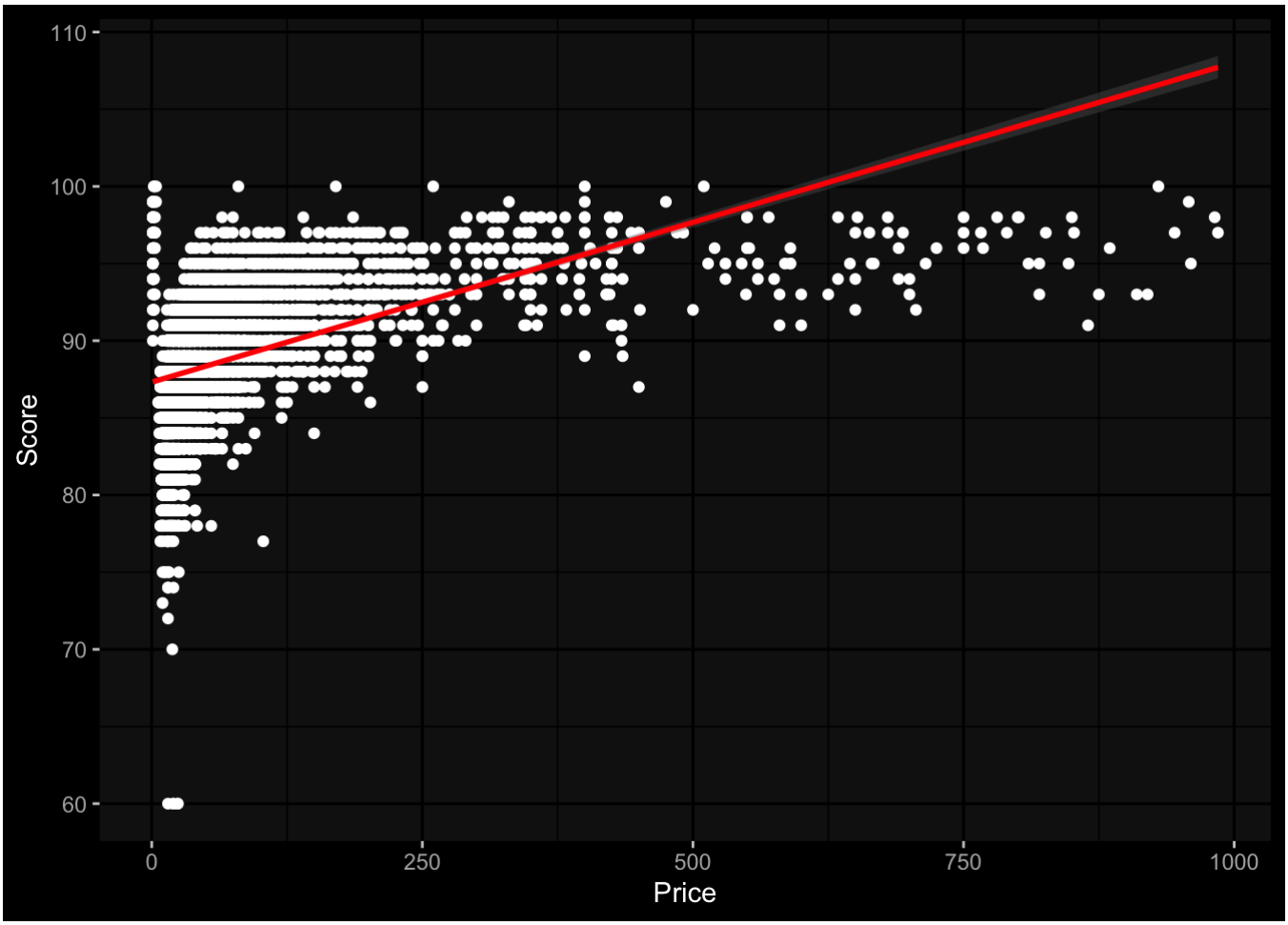
```
## Inverted geom defaults of fill and color/colour.
## To change them back, use invert_geom_defaults().
```

```
price_score_plot
```

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

```
## Warning: Removed 4663 rows containing non-finite values (stat_smooth).
```

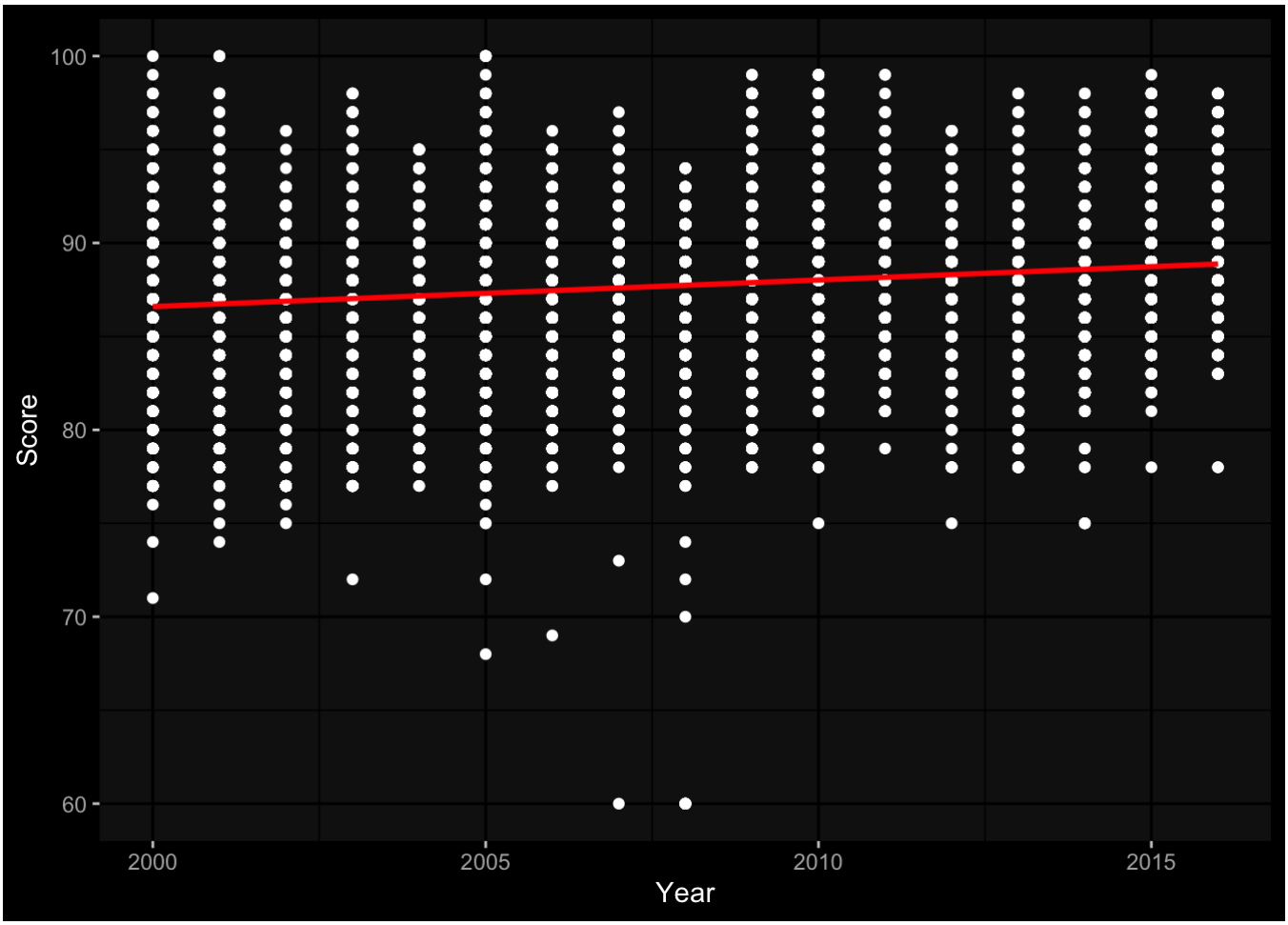
```
## Warning: Removed 4663 rows containing missing values (geom_point).
```




```
score_year_plot <- ggplot(wine_fixed, aes(x = Year, y = Score)) +
  geom_point() +
  geom_smooth(method = "lm", color = "red") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), # Turn of the background grid
    panel.grid.minor = element_blank(),
    panel.background = element_blank()) +
  dark_theme_gray()

score_year_plot
```

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

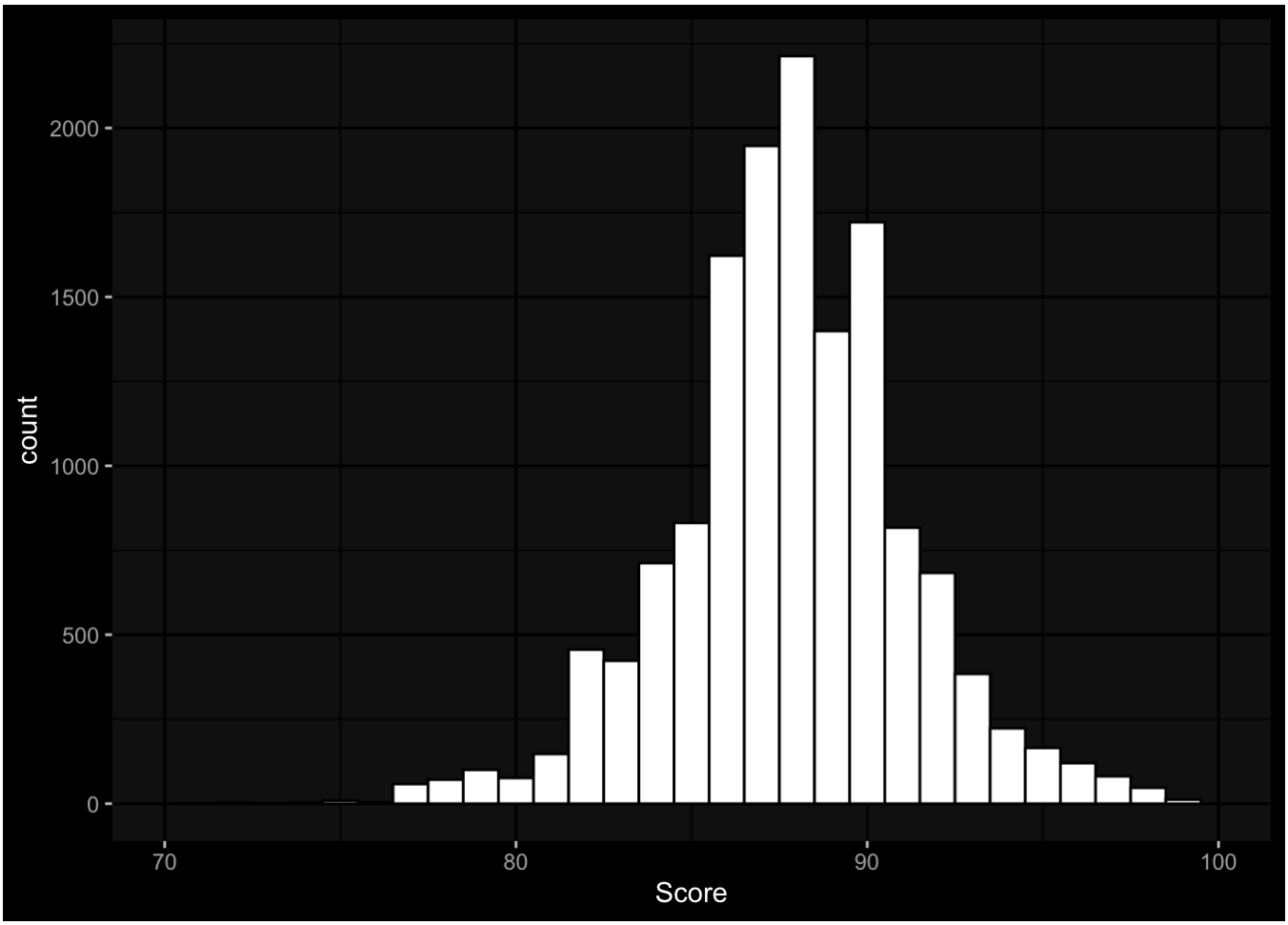



```
hist_score_plot <- ggplot(wine_fixed, aes(x = Score)) +
  geom_histogram(binwidth = 1, color="black", fill="white") +
  theme(panel.grid.major = element_blank(), # Turn of the background grid
    panel.grid.minor = element_blank(),
    panel.background = element_blank()) +
  xlim(70, 100) +
  dark_theme_gray()

hist_score_plot
```

```
## Warning: Removed 9 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



Linear regressions

```
lm_1 <- lm(Score ~ Price, data = wine_fixed)

summary(lm_1)
```

```
##
## Call:
## lm(formula = Score ~ Price, data = wine_fixed)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -27.8140  -1.6276   0.1031   1.9582  12.6416
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 8.732e+01  3.638e-02 2399.93   <2e-16 ***
## Price       2.071e-02  3.951e-04   52.42   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.015 on 9684 degrees of freedom
##   (4663 observations deleted due to missingness)
## Multiple R-squared:  0.2211, Adjusted R-squared:  0.221
## F-statistic:  2748 on 1 and 9684 DF,  p-value: < 2.2e-16
```

```
lm_2 <- lm(Score ~ Year, data = wine_fixed)

summary(lm_2)
```

```
##
## Call:
## lm(formula = Score ~ Year, data = wine_fixed)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -27.7350  -1.8741  -0.0231   2.1193  13.4040
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.982e+02  1.240e+01  -15.98   <2e-16 ***
## Year         1.424e-01  6.176e-03   23.05   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.406 on 14347 degrees of freedom
## Multiple R-squared:  0.03573,    Adjusted R-squared:  0.03566
## F-statistic: 531.5 on 1 and 14347 DF,  p-value: < 2.2e-16
```

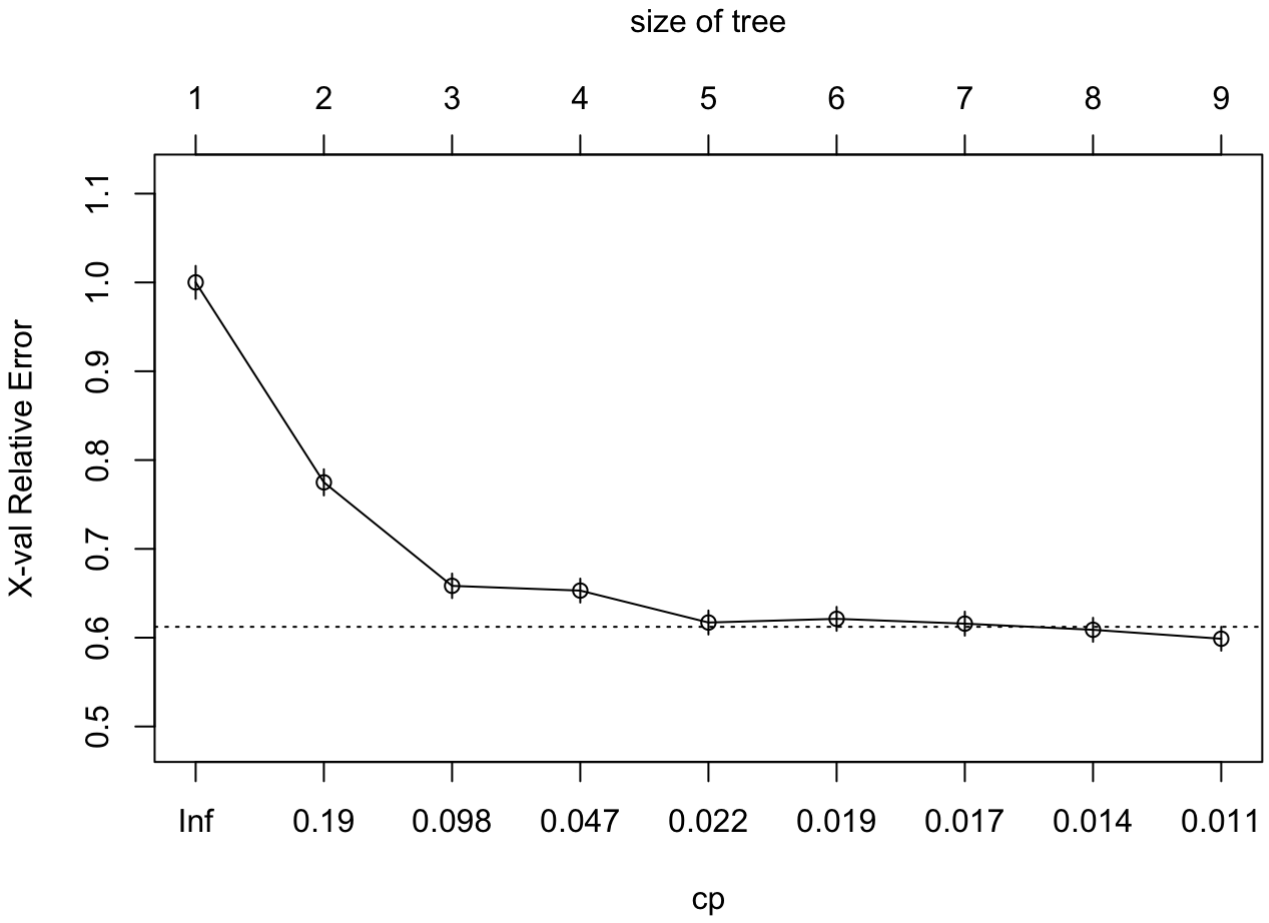
```
# Test glm with factors
# Doesnt work obviously
# log_fit_1 <- glm(Score ~., # Set formula
#              family=gaussian(link='identity'), # Set logistic regression
#              data= wine_fixed) # Set dataset
# summary(log_fit_1)
```

```
#summary(log_fit_1)
```

```
library(rpart)

tree_1 <- rpart(Score ~., # Set tree formula
                data = wine_fixed)

plotcp(tree_1) # Plot cp
```




```
tree_2 <- tree_1 <- rpart(Score ~., # Set tree formula
                          data = wine_fixed, # Set data
                          control = rpart.control(cp = 0.017)) # Set parameters
```

```
library(rattle)                  # Fancy tree plot
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```r
library(RColorBrewer)               # Color selection for fancy tree plot

summary(tree_2)
```