Worksheet 4 Group 1111

Ammara Akhtar, Afia Ibnath, and Reuben Walker

12 May 2024

```
library(tidyverse)
```

Load libraries

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(ggflags)
library(ggpubr)
library(latex2exp)
```

Data summary

```
chocolate_data <- read.csv('chocolate.csv')
print(chocolate_data)
```

```
##           Country Nobel.prizes.per.capita..scaled.by.10.million.
## 1     Switzerland                                          30.431
## 2         Austria                                          23.995
## 3         Ireland                                          14.572
## 4         Germany                                          13.124
## 5  United Kingdom                                          19.978
## 6          Sweden                                          30.052
## 7          Norway                                          24.284
## 8          Poland                                           3.149
## 9         Belgium                                           8.697
## 10        Finland                                           9.021
```

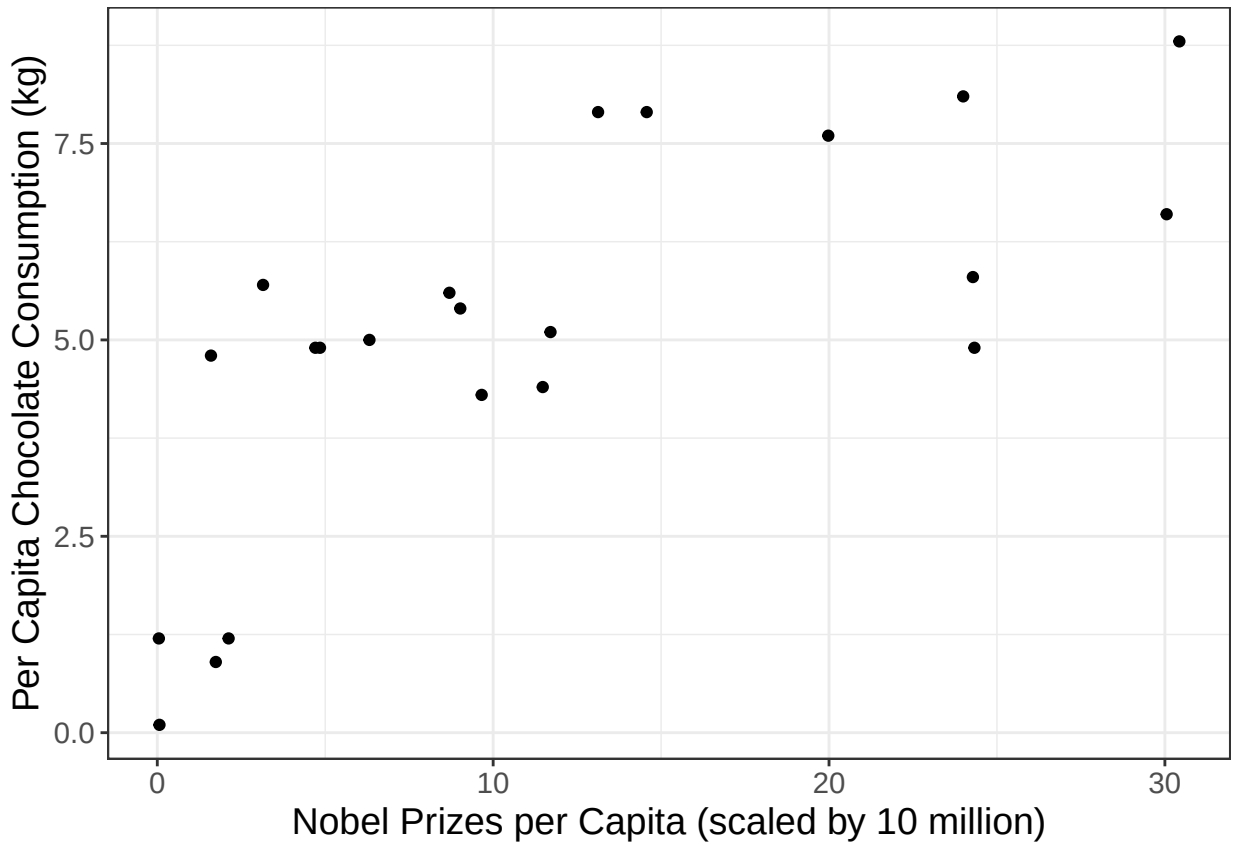
```
## 11    Netherlands                                          11.707
## 12    New Zealand                                           6.316
## 13        Denmark                                          24.329
## 14      Australia                                           4.844
## 15 Czech Republic                                           4.706
## 16         Russia                                           1.598
## 17  United States                                          11.476
## 18         France                                           9.658
## 19          Japan                                           2.123
## 20         Brazil                                           0.047
## 21   South Africa                                           1.742
## 22          China                                           0.064
##    Per.capita.chocolate.consumption..kg.
## 1                                    8.8
## 2                                    8.1
## 3                                    7.9
## 4                                    7.9
## 5                                    7.6
## 6                                    6.6
## 7                                    5.8
## 8                                    5.7
## 9                                    5.6
## 10                                   5.4
## 11                                   5.1
## 12                                   5.0
## 13                                   4.9
## 14                                   4.9
## 15                                   4.9
## 16                                   4.8
## 17                                   4.4
## 18                                   4.3
## 19                                   1.2
## 20                                   1.2
## 21                                   0.9
## 22                                   0.1
```

```r
# Summary statistics
summary(chocolate_data)
```

```
##    Country          Nobel.prizes.per.capita..scaled.by.10.million.
##  Length:22          Min.   : 0.047
##  Class :character   1st Qu.: 3.538
##  Mode  :character   Median : 9.339
##                     Mean   :11.632
##                     3rd Qu.:18.627
##                     Max.   :30.431
##  Per.capita.chocolate.consumption..kg.
##  Min.   :0.10
##  1st Qu.:4.50
##  Median :5.05
##  Mean   :5.05
##  3rd Qu.:6.40
##  Max.   :8.80
```

Visualize the dependent and independent variables

```
ggplot(chocolate_data, aes(x = Nobel.prizes.per.capita..scaled.by.10.million., y = Per.capita.chocolate
  geom_point() +
  labs(x = "Nobel Prizes per Capita (scaled by 10 million)", y = "Per Capita Chocolate Consumption (kg)
  theme_bw()+
  theme(text= element_text(size=14))
```




```
# Fit a simple linear regression model
lm_fit <- lm(Per.capita.chocolate.consumption..kg. ~ Nobel.prizes.per.capita..scaled.by.10.million., da

summary(lm_fit)$coefficients
```

```
##                                              Estimate Std. Error  t value
## (Intercept)                                 2.9969925 0.57883419 5.177636
## Nobel.prizes.per.capita..scaled.by.10.million. 0.1764903 0.03841546 4.594253
##                                                 Pr(>|t|)
## (Intercept)                                 4.575076e-05
## Nobel.prizes.per.capita..scaled.by.10.million. 1.756772e-04
```

The analysis suggests that there is a statistically significant relationship between Nobel prizes per capita scaled by 10 million and Per Capita Chocolate Consumption. For every one-unit increase in Nobel prizes per capita scaled by 10 million, there is an expected increase in Per Capita Chocolate Consumption by approximately 0.1765 units.

Compute intercept, slope, and R-squared

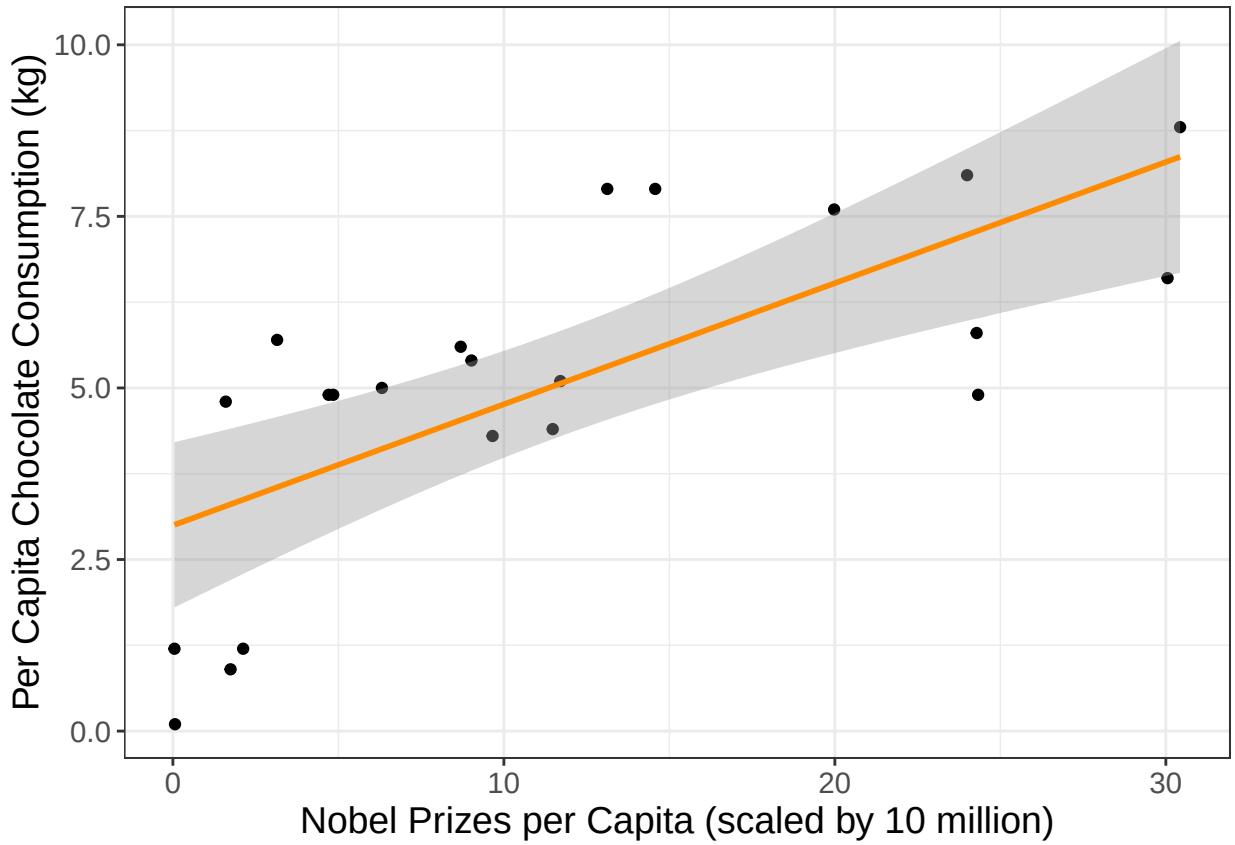
```
intercept <- coef(lm_fit)[1]
slope <- coef(lm_fit)[2]
r_squared <- summary(lm_fit)$r.squared
cat("Intercept: ", intercept, "\n",
    "Slope: ", slope, "\n",
    "R_squared: ", r_squared, "\n")
```

```
## Intercept:  2.996993
##  Slope:  0.1764903
##  R_squared:  0.5134667
```

Add regression line using geom_smooth

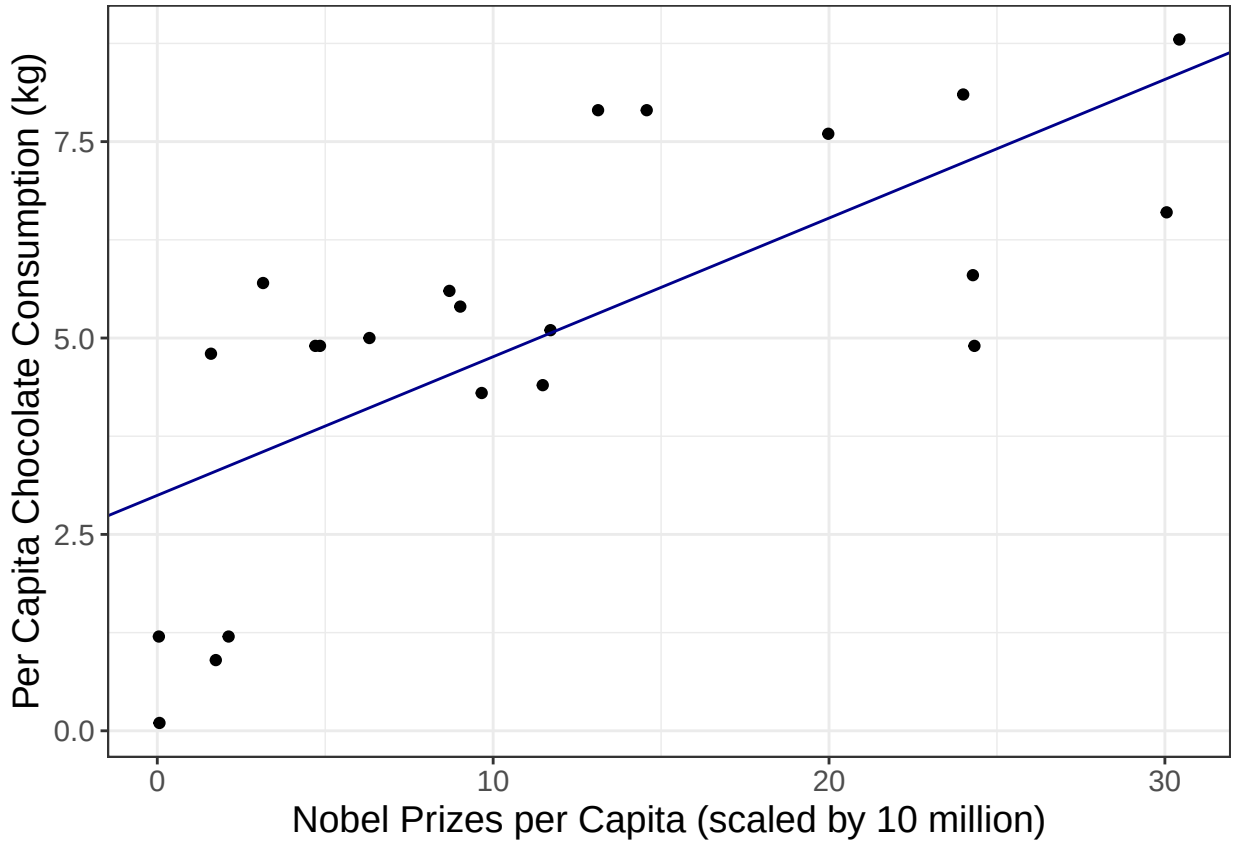
```
ggplot(chocolate_data, aes(x = Nobel.prizes.per.capita..scaled.by.10.million., y = Per.capita.chocolate
  geom_point() +
  geom_smooth(method = "lm", color = "darkorange") +
  labs(x = "Nobel Prizes per Capita (scaled by 10 million)", y = "Per Capita Chocolate Consumption (kg)
  theme_bw()+
  theme(text= element_text(size=14))
```

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



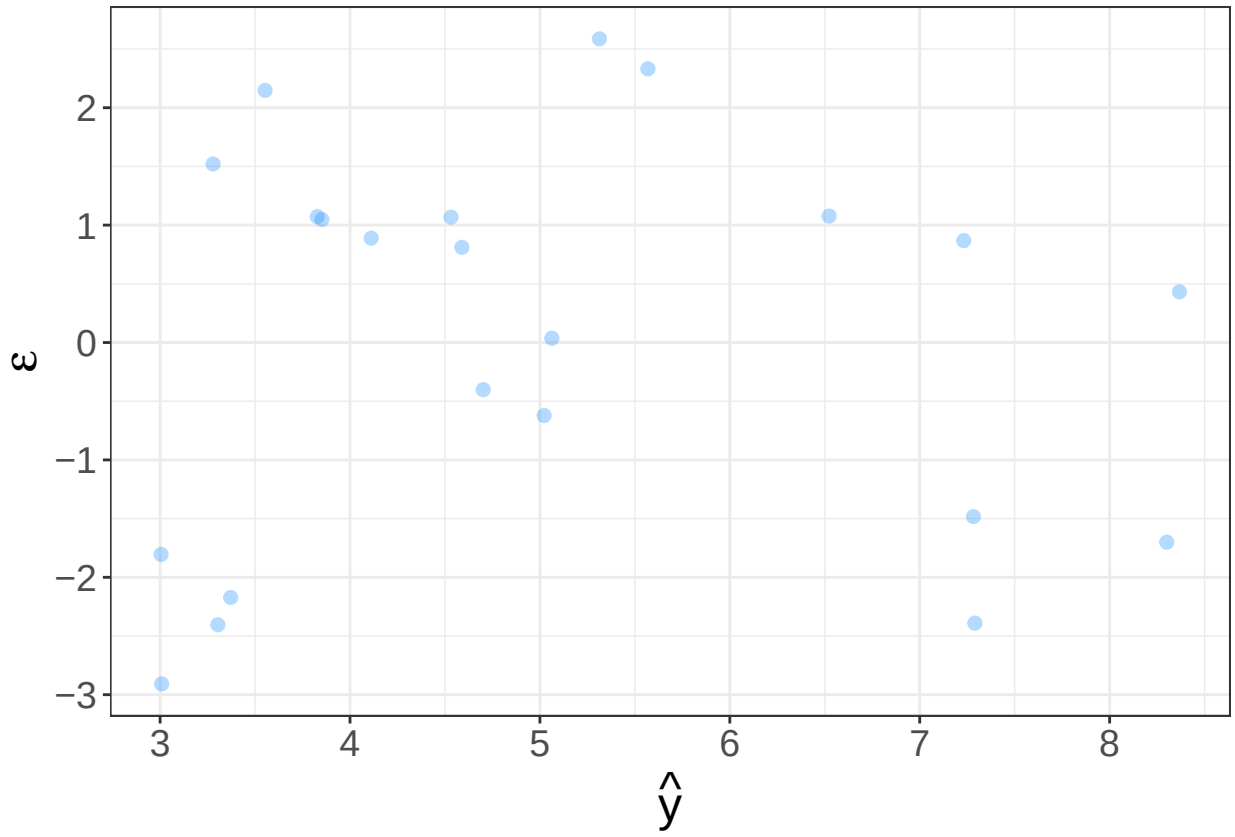
Add regression line manually

```
ggplot(chocolate_data, aes(x = Nobel.prizes.per.capita..scaled.by.10.million., y = Per.capita.chocolate
  geom_point() +
  geom_abline(intercept = intercept, slope = slope, color = "darkblue") +
  labs(x = "Nobel Prizes per Capita (scaled by 10 million)", y = "Per Capita Chocolate Consumption (kg)
  theme_bw()+
  theme(text= element_text(size=14))
```



Check assumptions with visualizations

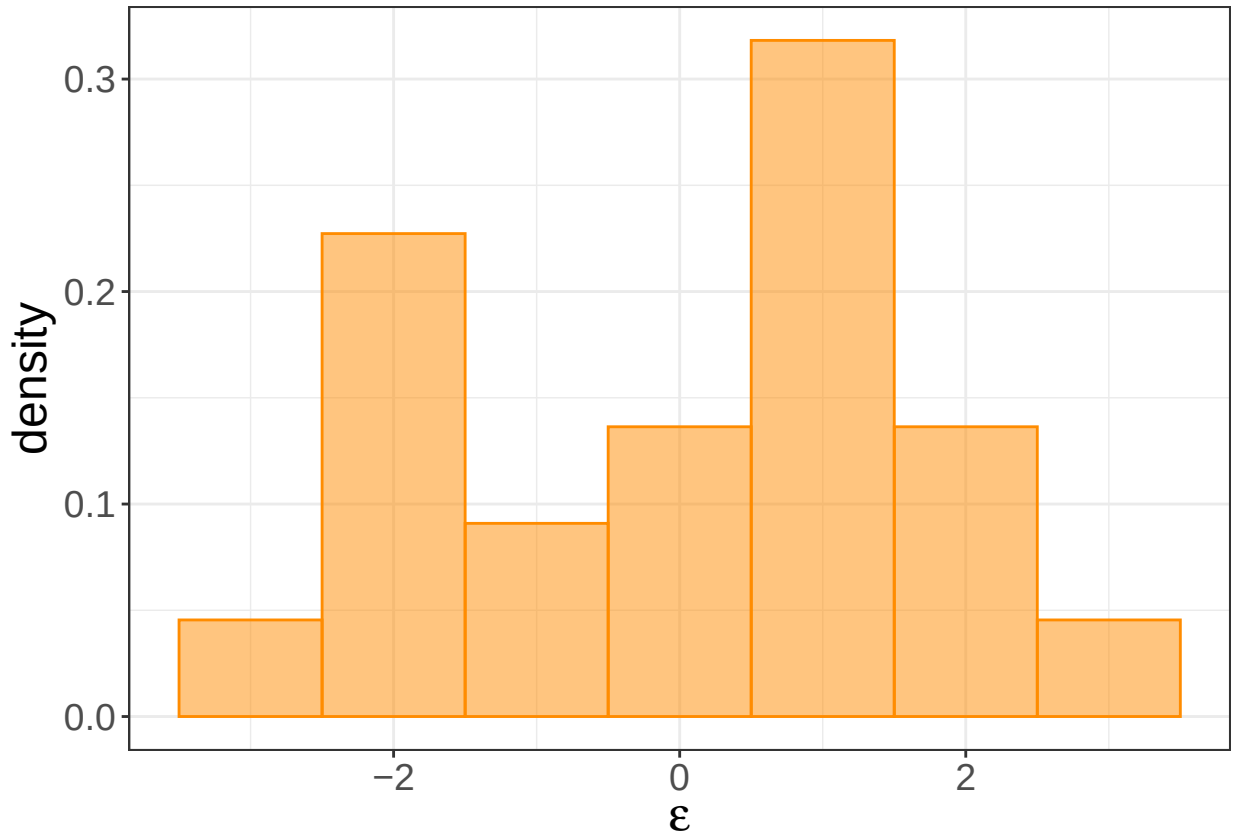
```
ggplot(lm_fit, aes(x = .fitted, y = .resid)) +
geom_point (colour = "dodgerblue", size = 2, alpha = 0.33) +
xlab (expression (hat(y))) + ylab (expression (epsilon)) +
theme_bw() + theme (text = element_text(size = 18))
```



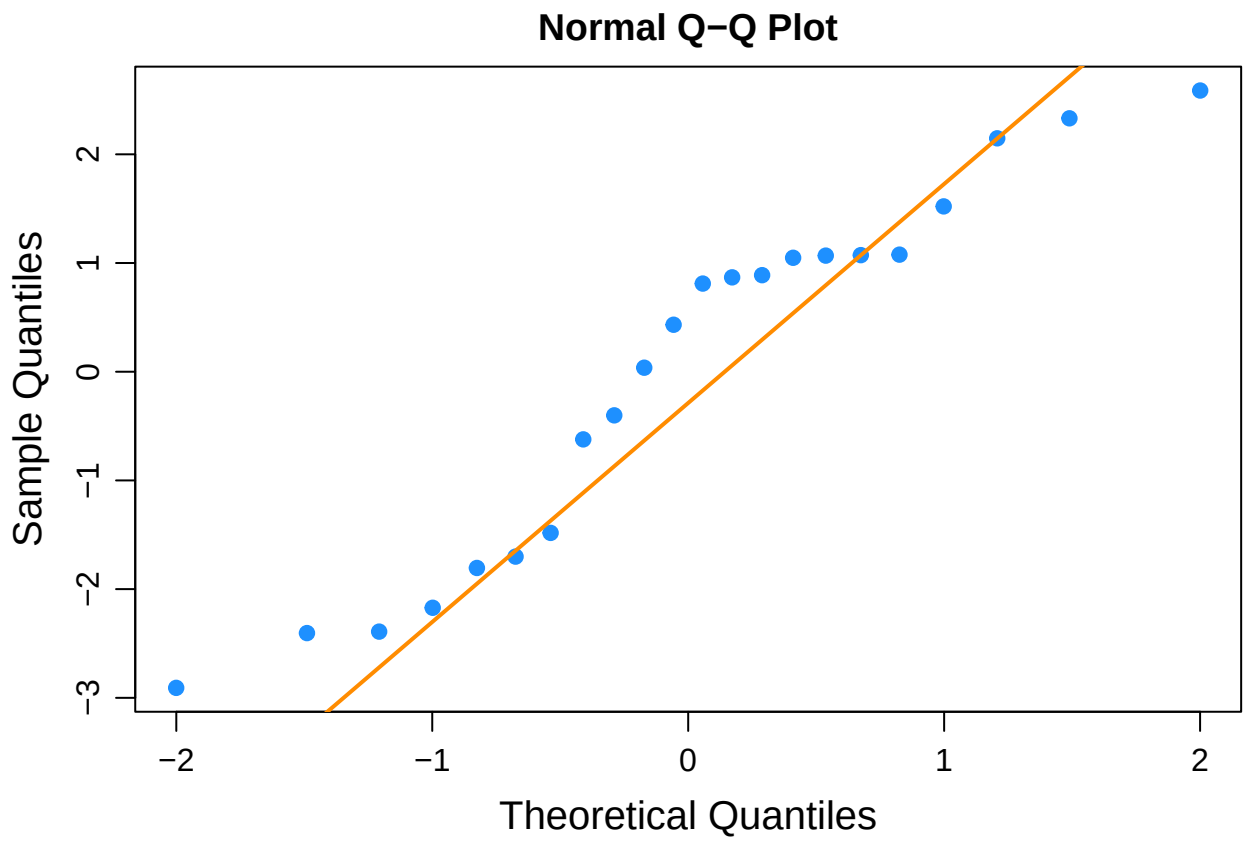

```
ggplot(lm_fit, aes(x = .resid, y = ..density..)) +
theme_bw() + theme (text = element_text(size = 18)) + xlab (expression(epsilon)) + geom_histogram (binw
```

The residuals are randomly scattered around zero which means equal variance assumption is not violated. But, the sample size is apparantely lower. So, a larger sample size might generate different results.

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
par(mar = c(3.6, 3.6, 2.1, 0.1))
par(mgp = c(2.25, 0.75, 0))
qqnorm(lm_fit$residuals, col = "dodgerblue", cex.lab = 1.25, pch = 19)
qqline(lm_fit$residuals, col = "darkorange", lwd = 2)
```



Analysing the Q-Q plot we can observe some non-normality in the residuals. There might be outliers in the data. Which means assumptions of our model is violated.

Question 2

```
library(tidyverse)
four_df <- read.csv('four_datasets.csv')
str(four_df)
```

```
## 'data.frame':    44 obs. of  3 variables:
##  $ Dataset: chr  "A" "A" "A" "A" ...
##  $ x      : int  10 8 13 9 11 14 6 4 12 7 ...
##  $ y      : num  8.04 6.95 7.58 8.81 8.33 ...
```

```
four_df$Dataset
```

```
##  [1] "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "B" "B" "B" "B" "B" "B" "B" "B"
## [20] "B" "B" "B" "C" "C" "C" "C" "C" "C" "C" "C" "C" "C" "D" "D" "D" "D" "D"
## [39] "D" "D" "D" "D" "D" "D"
```

```
group <- four_df %>% group_by(Dataset)
group %>% summarise(
  x_mean = mean(x),
  #x_sd = sd(x),
  y_mean = mean(y),
  #y_sd = sd(y)
)
```

```
## # A tibble: 4 x 3
##   Dataset x_mean y_mean
##   <chr>    <dbl>  <dbl>
## 1 A            9   7.50
## 2 B            9   7.50
## 3 C            9   7.5
## 4 D            9   7.50
```

```
group %>% summarise(
  #x = mean(x),
  x_sd = sd(x),
  #y = mean(y),
  y_sd = sd(y)
)
```

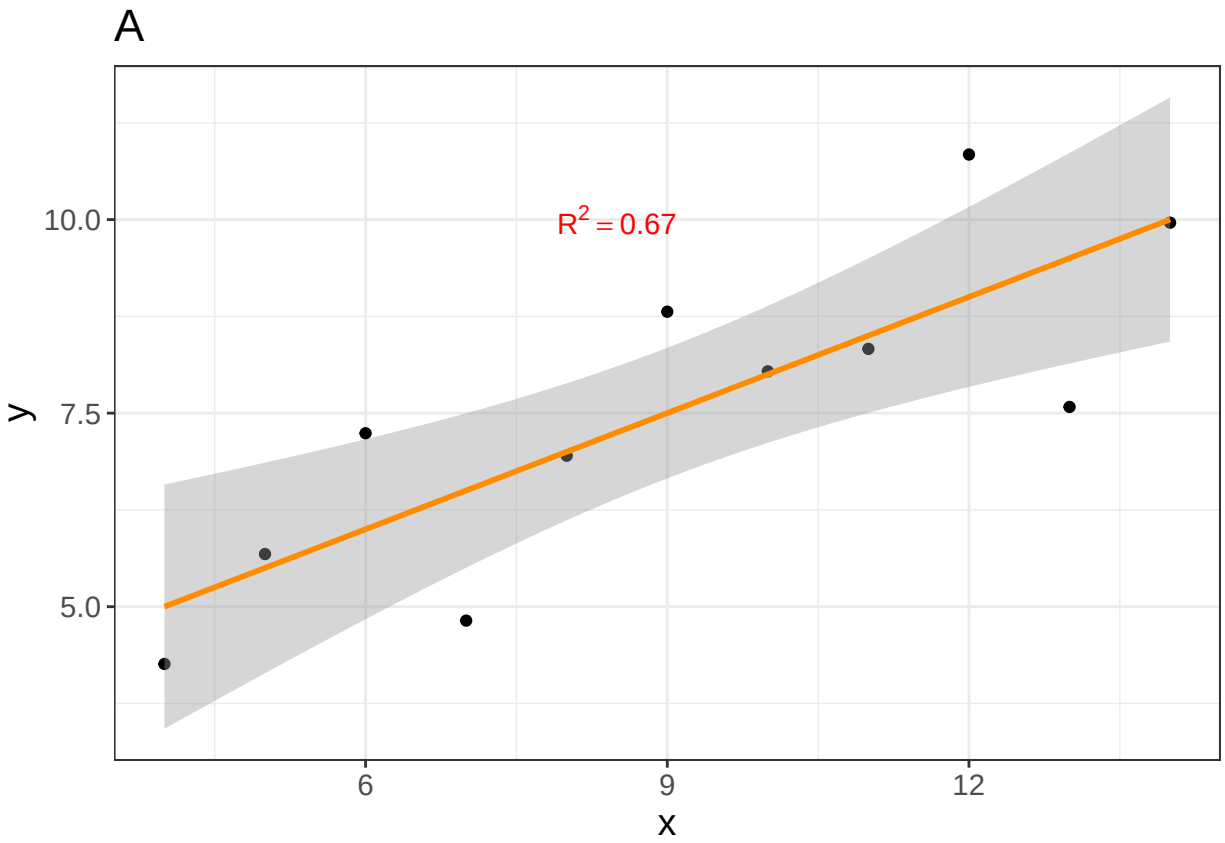
```
## # A tibble: 4 x 3
##   Dataset  x_sd  y_sd
##   <chr>   <dbl> <dbl>
## 1 A        3.32  2.03
## 2 B        3.32  2.03
## 3 C        3.32  2.03
## 4 D        3.32  2.03
```

Looking at our descriptive statistics, the groupings have essentially identical means and standard deviations.

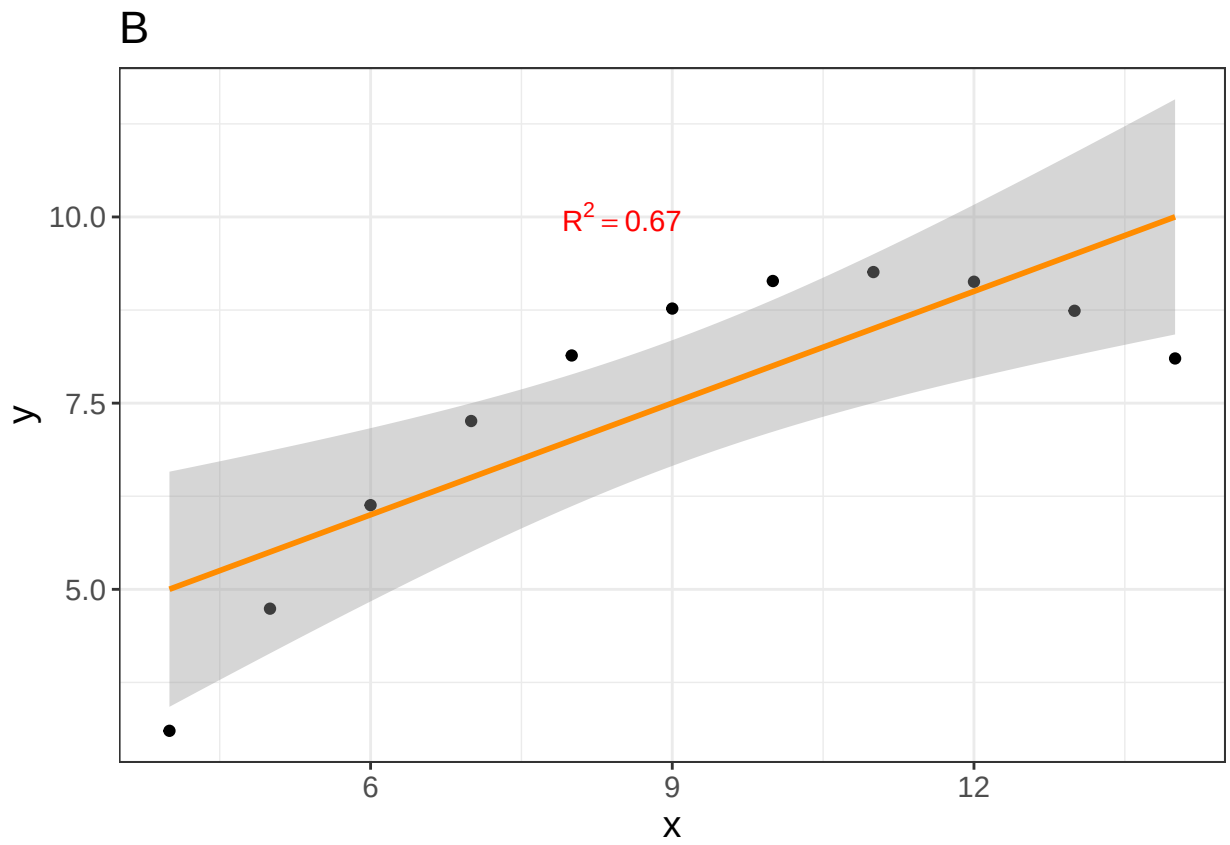
```
groups <- list("A","B","C","D")
for (i in 1:length(groups)){
  group_i <- groups[i]
  df_i <- subset(four_df, four_df$Dataset == group_i)
  lm_fit_i <- lm(y ~ x, data = df_i)

  #plot(d$x, d$y)
  print(ggplot(df_i, aes(x = x, y = y)) +
  geom_point() +
  annotate(geom="text", x=8.5, y=10, label=paste("R^2== ",round(summary(lm_fit_i)$r.squared, digits=2))
  geom_smooth(method = "lm", color = "darkorange") +
  labs(x = "x", y = "y", title=group_i) +
  theme_bw()+
  theme(text= element_text(size=14)))
}
```

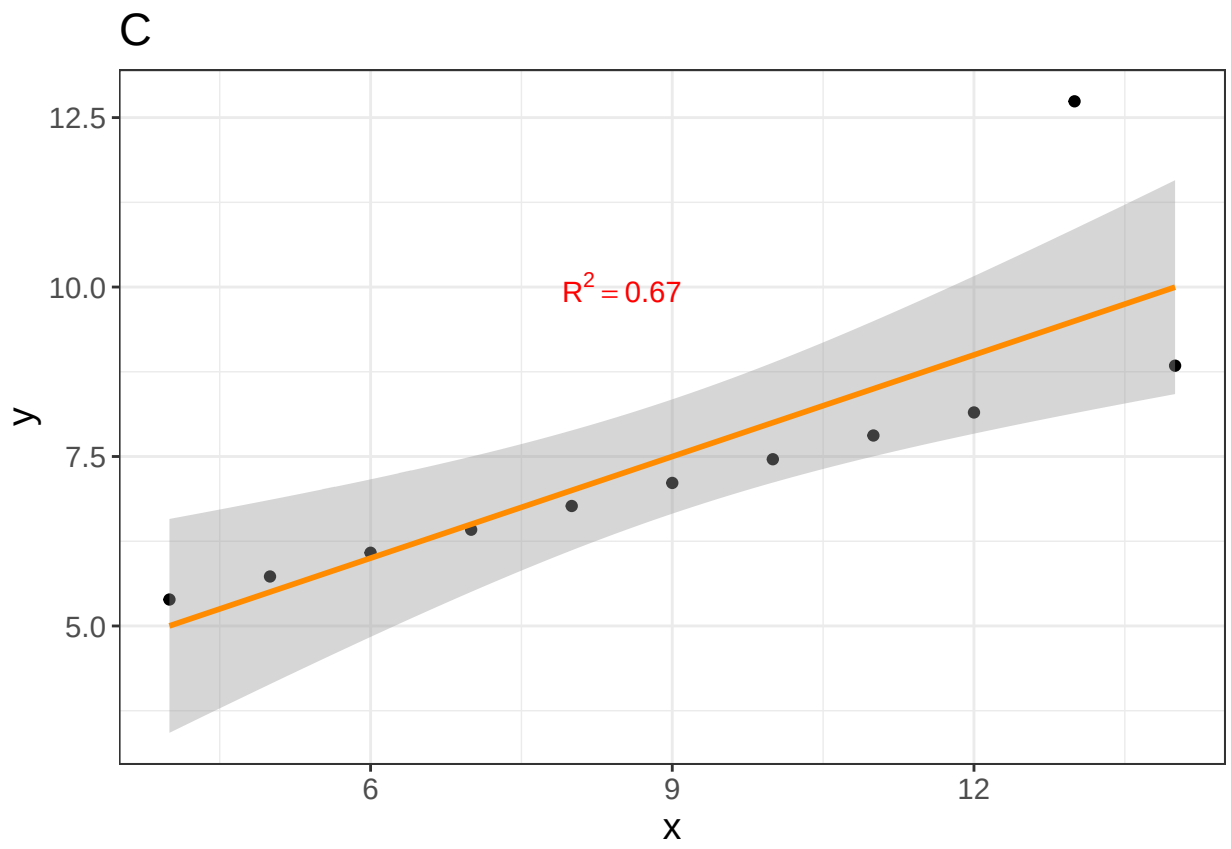
```
## `geom_smooth()` using formula = 'y ~ x'
```



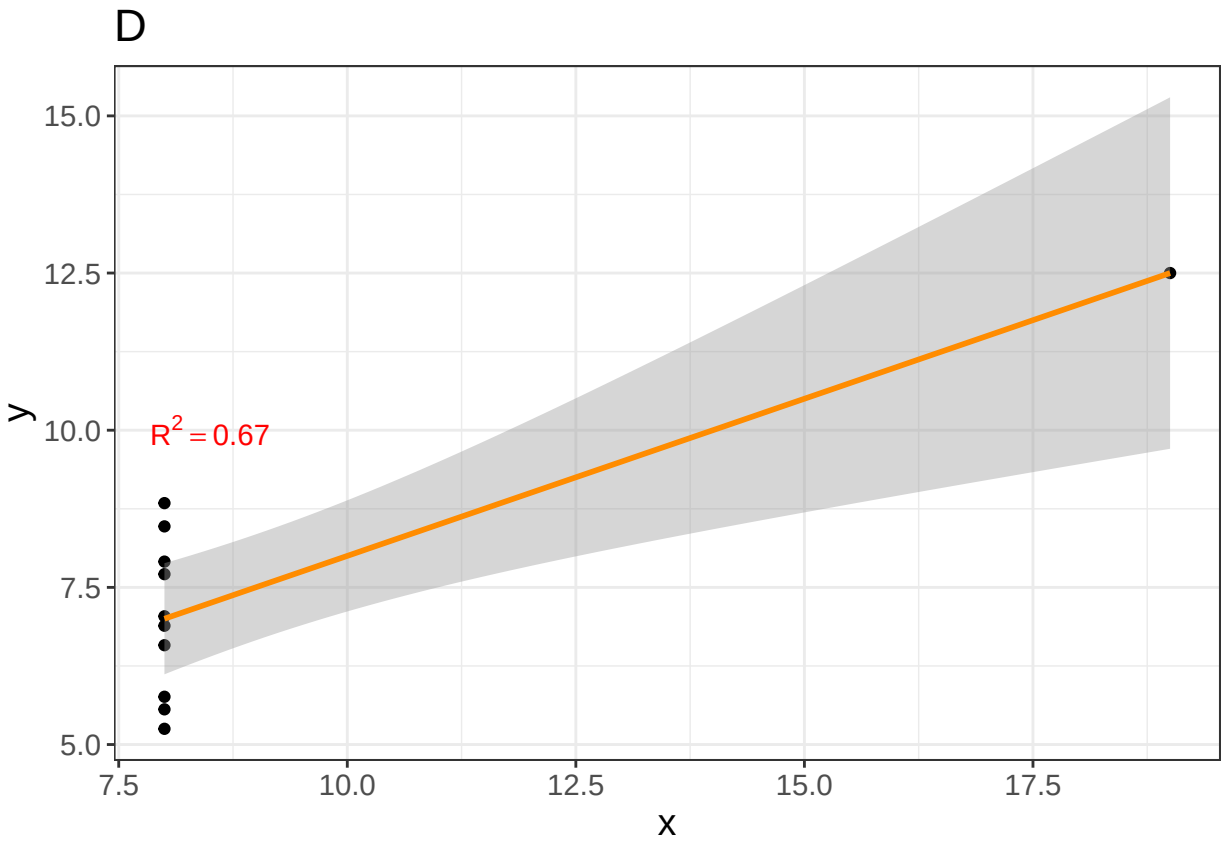

```
## `geom_smooth()` using formula = 'y ~ x'
```




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

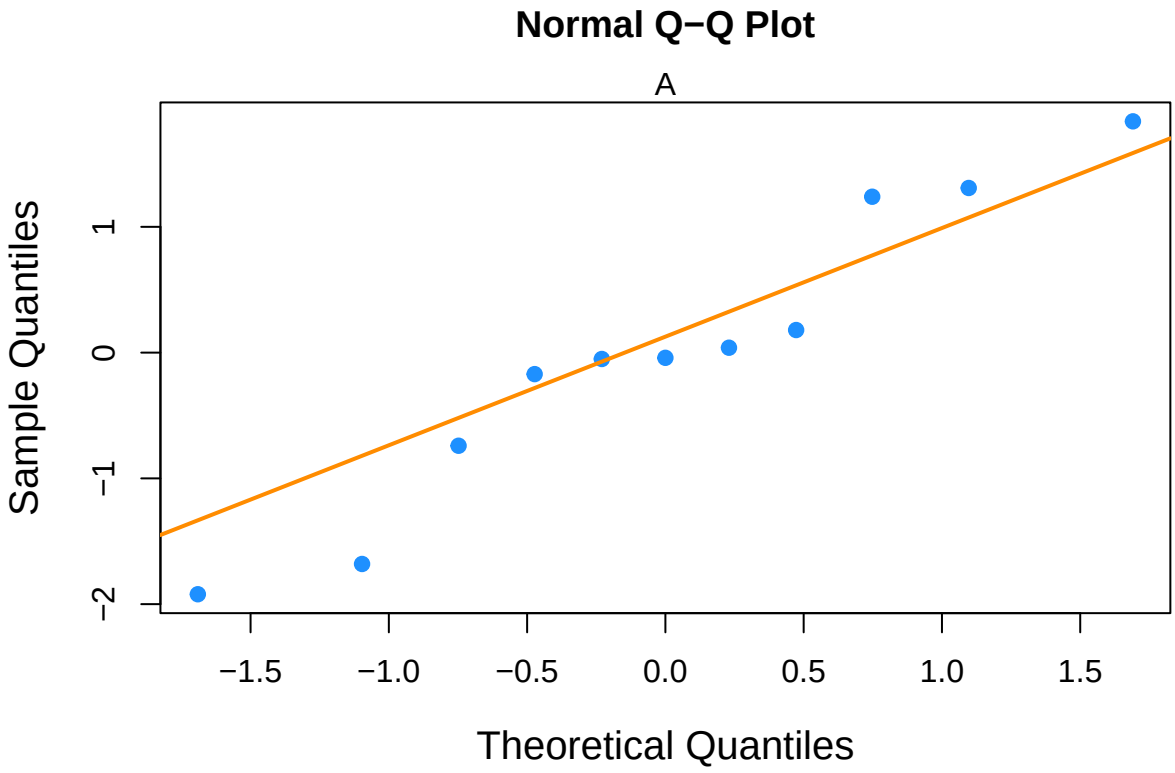


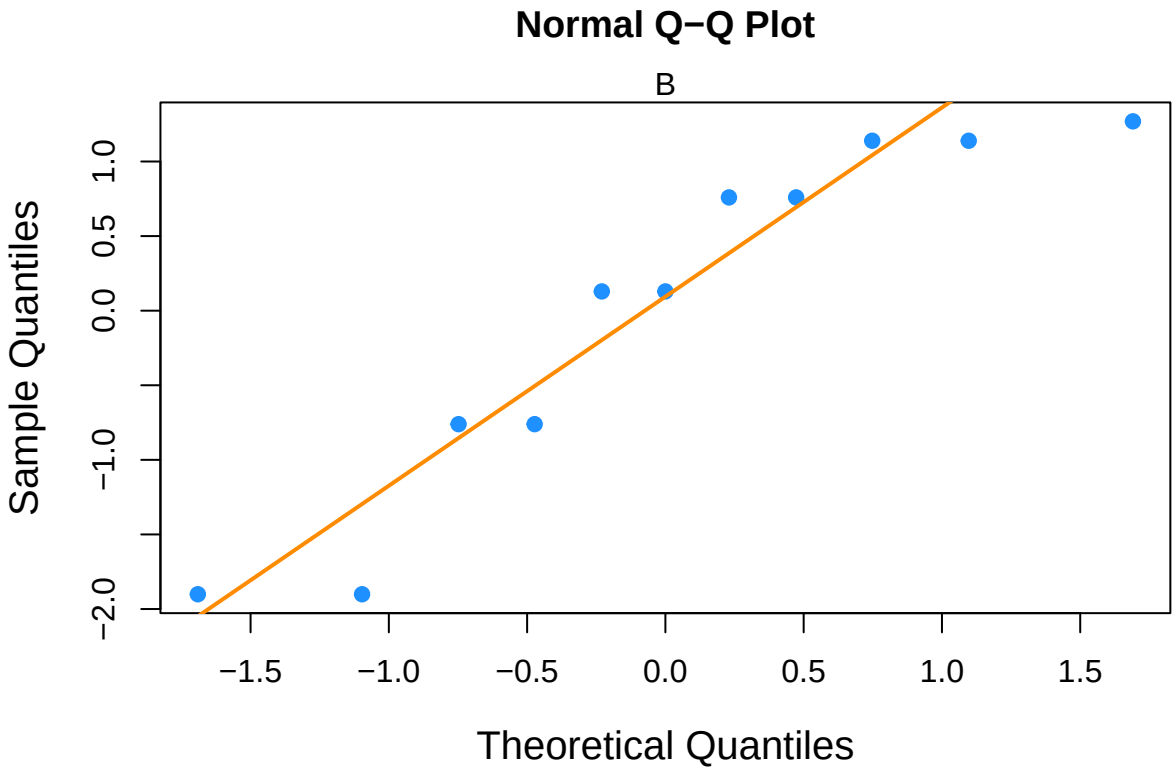

```
## `geom_smooth()` using formula = 'y ~ x'
```

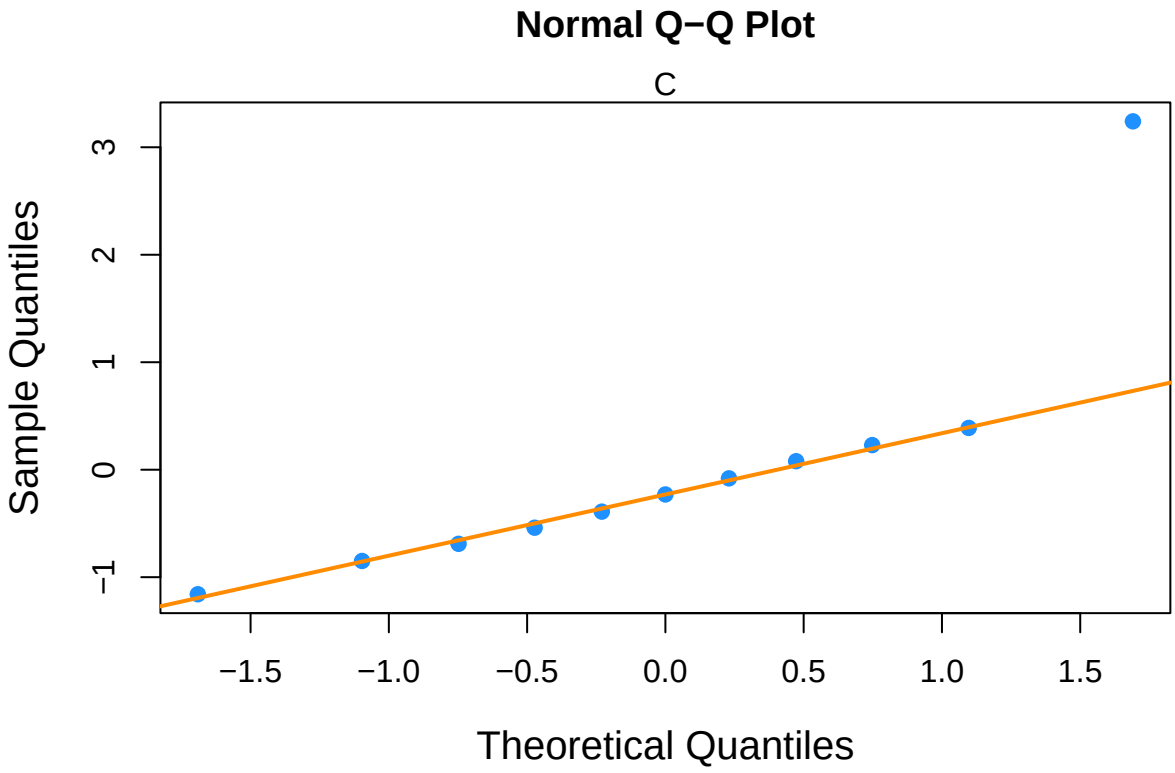


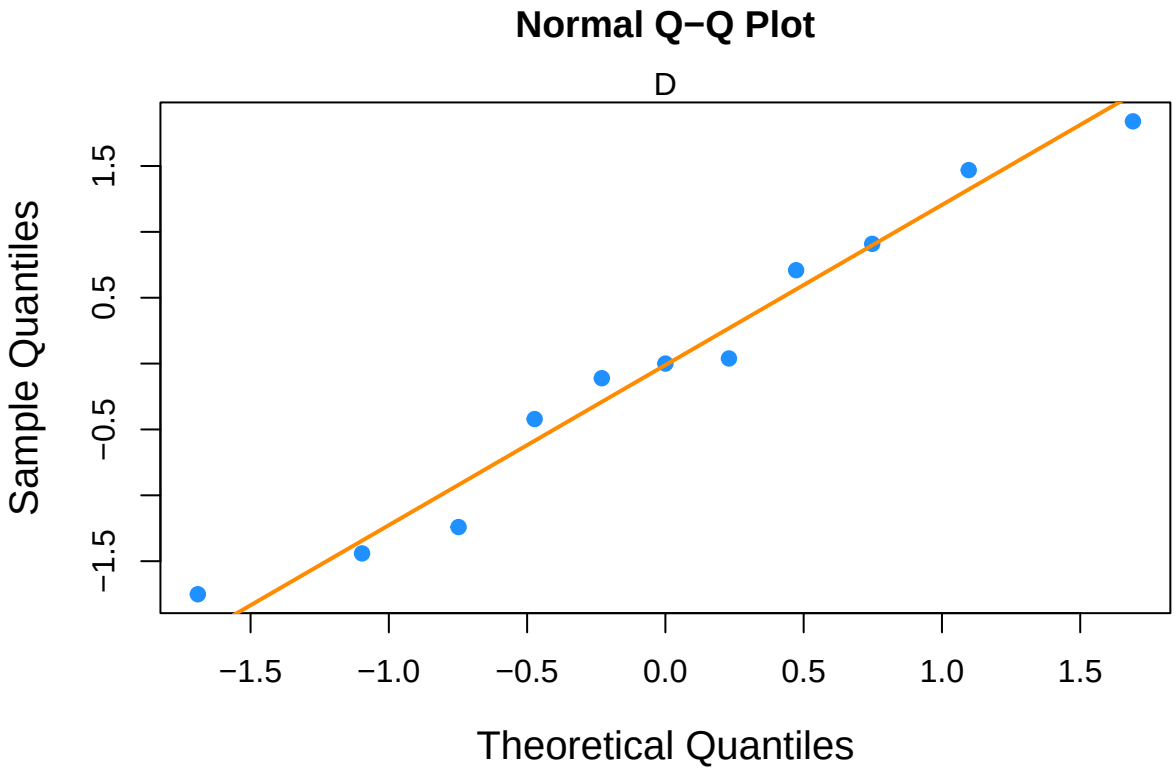
For all four groupings, we have essentially identical descriptive statistics and correlation coefficients. However, we have data with a nonlinear relationship (B), a single outlier in a clear linear relationship (C), and data with a clear outlier amongst a population with zero variation (D).

```
groups <- list("A","B","C","D")
for (i in 1:length(groups)){
  group_i <- groups[i]
  df_i <- subset(four_df, four_df$Dataset == group_i)
  lm_fit_i <- lm(y ~ x, data = df_i)
  qqnorm(lm_fit_i$residuals, col = "dodgerblue", cex.lab = 1.25, pch = 19)
  qqline(lm_fit_i$residuals, col = "darkorange", lwd = 2)
  mtext(groups[i])
}
```


Normal Q−Q Plot
A
Sample Quantiles
Theoretical Quantiles
−2
−1
0
1
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


Normal Q−Q Plot
B
Sample Quantiles
Theoretical Quantiles
−2.0
−1.0
0.0
0.5
1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


Normal Q−Q Plot
C
Sample Quantiles
Theoretical Quantiles
−1
0
1
2
3
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5



Surprisingly, the QQ plots aren't that different, with the only true recognizable differences being the repeated stepwise patterns in the quadratic "B" data and the clear outlier in "C". This emphasizes the importance of visualizing your data prior to regression analysis.

Question 3

```
library(HistData)
str(GaltonFamilies)
```

```
## 'data.frame':    934 obs. of  8 variables:
##  $ family         : Factor w/ 205 levels "001","002","003",..: 1 1 1 1 2 2 2 2 3 3 ...
##  $ father         : num  78.5 78.5 78.5 78.5 75.5 75.5 75.5 75.5 75 75 ...
##  $ mother         : num  67 67 67 67 66.5 66.5 66.5 66.5 64 64 ...
##  $ midparentHeight: num  75.4 75.4 75.4 75.4 73.7 ...
##  $ children       : int  4 4 4 4 4 4 4 4 2 2 ...
##  $ childNum       : int  1 2 3 4 1 2 3 4 1 2 ...
##  $ gender         : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 1 1 2 1 ...
##  $ childHeight    : num  73.2 69.2 69 69 73.5 72.5 65.5 65.5 71 68 ...
```

```
#df_sons
```

We have a data frame with the height of the child and the height of their parents. Let's take the subset of sons.

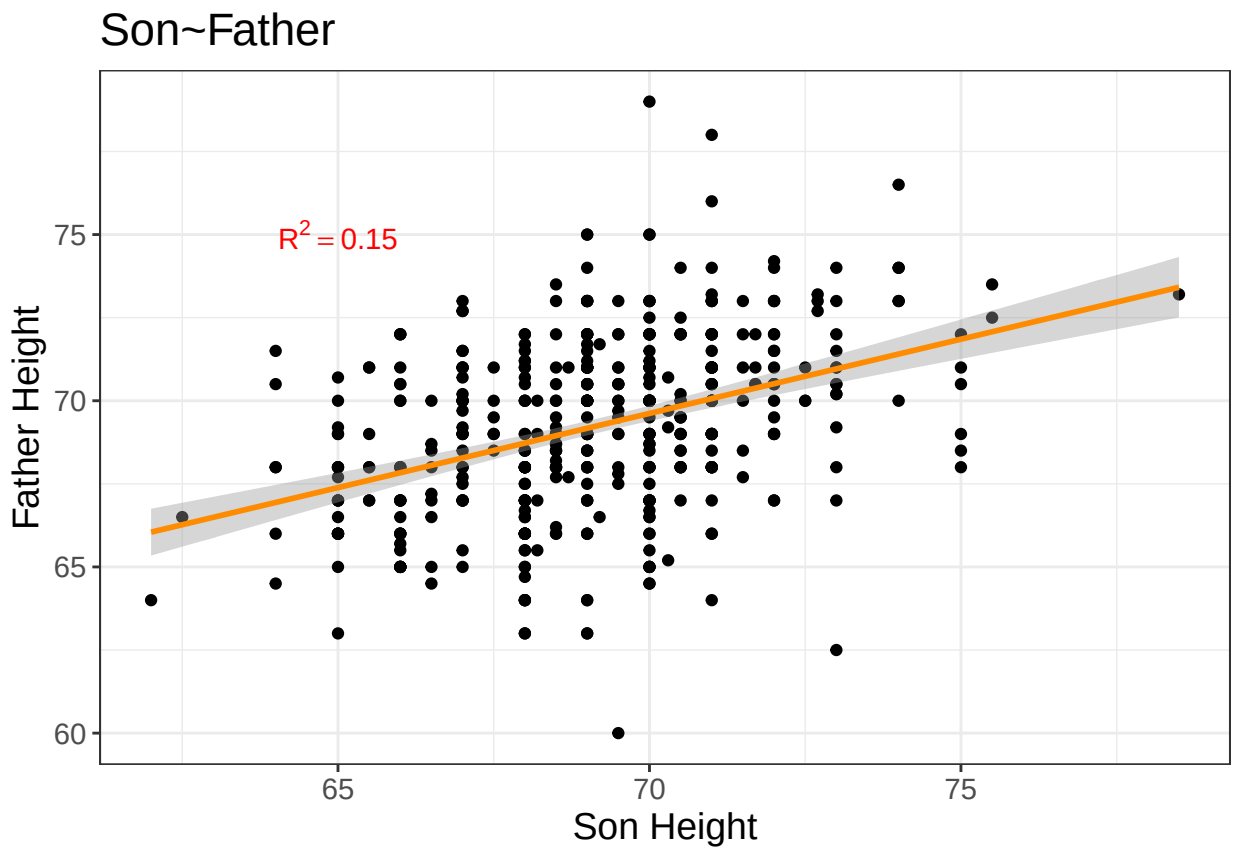
```r
df_sons = subset(GaltonFamilies,GaltonFamilies$gender == "male")
lm_fit_s <- lm(childHeight ~ father, data = df_sons)

summary(lm_fit_s)$coefficients
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 38.3625810 3.30837361 11.595601 1.425209e-27
## father       0.4465226 0.04782546  9.336504 3.737109e-19
```

```r
ggplot(df_sons, aes(x = father, y = childHeight)) +
  geom_point() +
  geom_smooth(method = "lm", color = "darkorange") +
  labs(x = "Son Height", y = "Father Height",
       title="Son~Father") +
  theme_bw()+
  annotate(geom="text", x=65, y=75, label=paste("R^2== ",round(summary(lm_fit_s)$r.squared, digits=2)),
  theme(text= element_text(size=14))
```

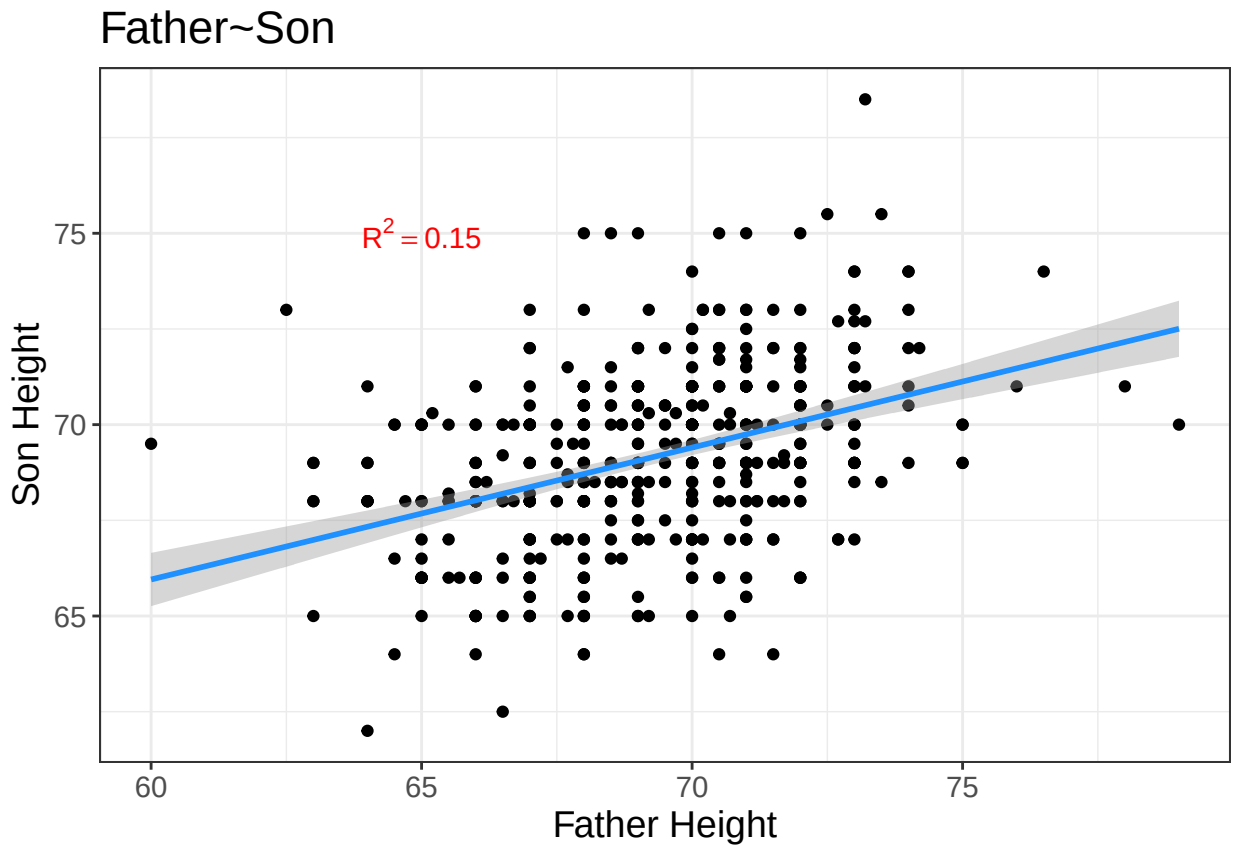
```
## `geom_smooth()` using formula = 'y ~ x'
```



Though there is a significant correlation between son height and father height, the R^2 value is only 0.15, meaning that 15% of the variation in son height can be explained through the height of the father. That means that 85% of the variation is unexplained by height of the father. If you wanted to estimate son height based on father height, a healthy portion of your prediction will be the mean height of the sample.

```
lm_fit_f <- lm(father ~ childHeight, data = df_sons)
ggplot(df_sons, aes(x = childHeight, y = father)) +
  geom_point() +
  geom_smooth(method = "lm", color = "dodgerblue") +
  labs(x = "Father Height", y = "Son Height",
       title="Father~Son") +
  theme_bw()+
  annotate(geom="text", x=65, y=75, label=paste("R^2== ",round(summary(lm_fit_f)$r.squared, digits=2)),
  theme(text= element_text(size=14))
```

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




```
summary(lm_fit_s)$coefficients
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 38.3625810 3.30837361 11.595601 1.425209e-27
## father       0.4465226 0.04782546  9.336504 3.737109e-19
```

```
summary(lm_fit_f)$coefficients
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 45.2651223 2.55873194 17.690451 2.909078e-54
## childHeight  0.3448085 0.03693123  9.336504 3.737109e-19
```

```r
cat("Pearson's correlation: ", cor(df_sons$father, df_sons$childHeight))
```

```
## Pearson's correlation:  0.3923835
```

The slope of the regression line for the father~sonHeight is 0.34 and the slope of the regression line for son-Height~father is 0.45. The R^2 value for both is identical, showing that the Pearson's coefficient (calculated with mean-centered and standardized scale is identical at 0.39).

```r
summary(lm_fit_s)$coefficients
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 38.3625810 3.30837361 11.595601 1.425209e-27
## father       0.4465226 0.04782546  9.336504 3.737109e-19
```

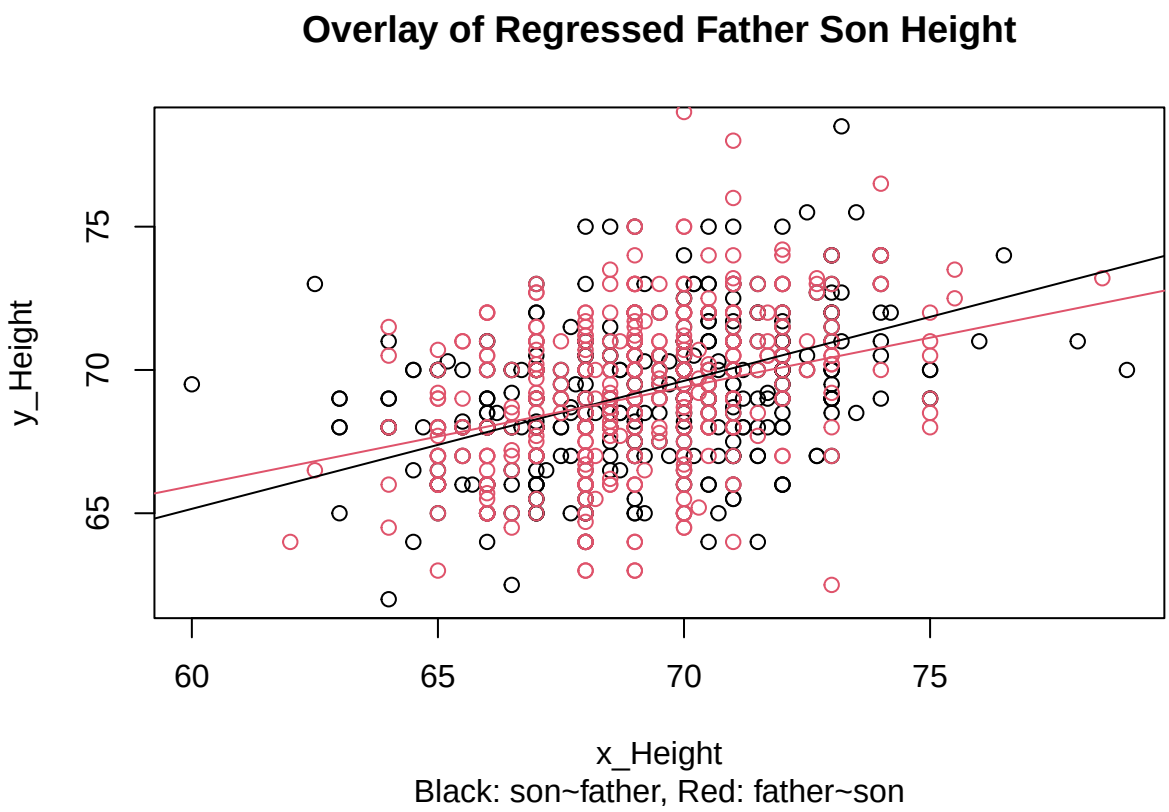
```r
summary(lm_fit_f)$coefficients
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 45.2651223 2.55873194 17.690451 2.909078e-54
## childHeight  0.3448085 0.03693123  9.336504 3.737109e-19
```

```r
cat("Pearson's correlation: ", cor(df_sons$father, df_sons$childHeight))
```

```
## Pearson's correlation:  0.3923835
```

```r
a_s <- summary(lm_fit_s)$coefficients[1]
b_s <- summary(lm_fit_s)$coefficients[2]
a_f <- summary(lm_fit_f)$coefficients[1]
b_f <- summary(lm_fit_f)$coefficients[2]
#ggplot(df_sons, aes(x = childHeight, y = father)) +
#  geom_point() +
#  #geom_smooth(method = "lm", color = "darkorange") +
#  labs(x = "x Height", y = "y Height") +
#  theme_bw()+
#  theme(text= element_text(size=14)) +
#  geom_abline(intercept=a_s, slope=b_s) +
#  geom_abline(intercept=a_f, slope=b_f)
plot(x=df_sons$childHeight, y=df_sons$father,
     xlab="x_Height", ylab="y_Height",
     main="Overlay of Regressed Father Son Height",
     sub="Black: son~father, Red: father~son")
points(x=df_sons$father, y=df_sons$childHeight, col=2)
abline(a=a_s, b=b_s)
abline(a=a_f, b=b_f, col=2)
```



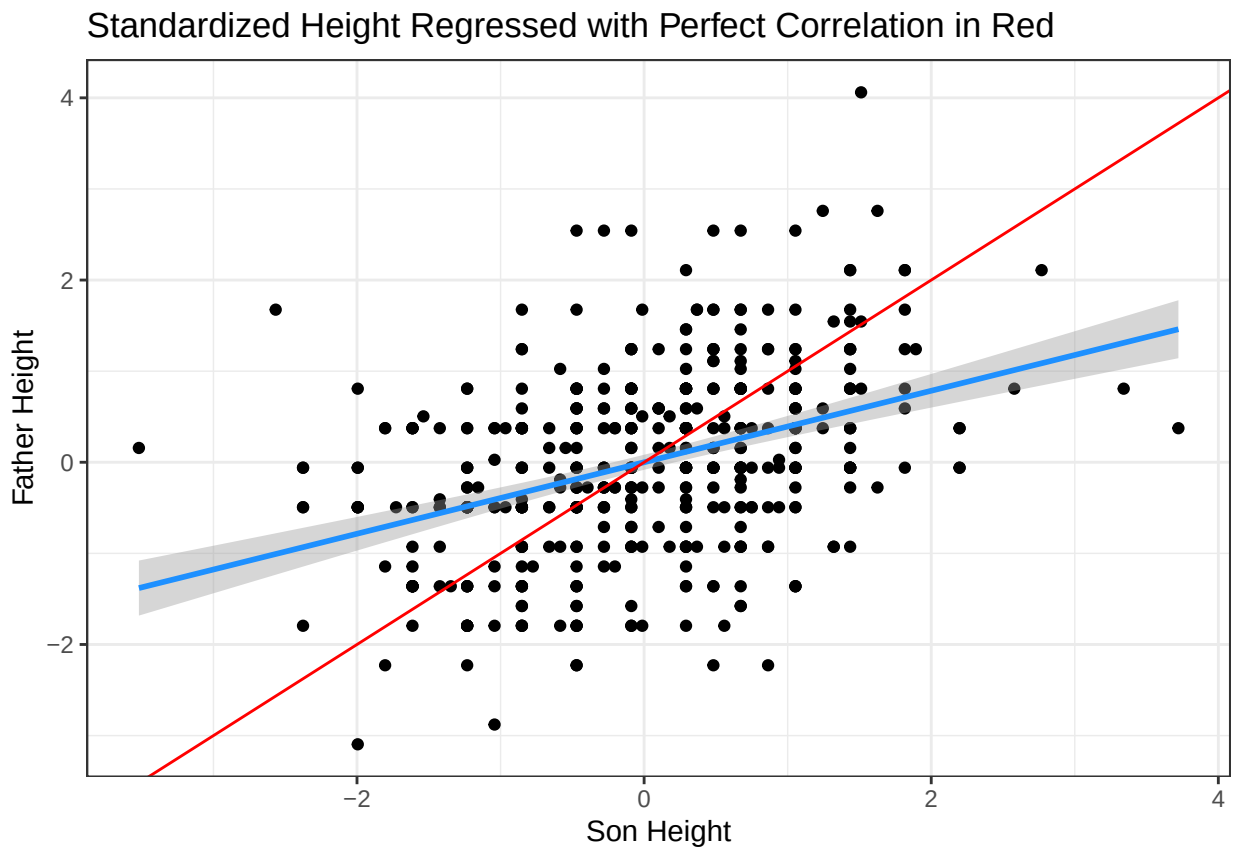

```
#x_means <- c(mean(df_sons$father),)
#abline(a=(mean(df$father)-mean(df$childHeight)), b=1, col=3)
#points(x=mean(df_sons$father),y=mean(df_sons$childHeight),pch=19)
#points(y=mean(df_sons$father),x=mean(df_sons$childHeight),pch=19,col=2)
```

So for both sons and fathers, an extreme case is likely to be paired with a less extreme case. A very tall father's child will likely be less tall than him and a very tall child is unlikely to have come from a very tall father.

Let's look at the scaled data:

```
df_sons$scaled_f <- scale(df_sons$father)
df_sons$scaled_s <-scale(df_sons$childHeight)
lm_fit_f <- lm(scaled_f ~ scaled_s, data = df_sons)
ggplot(df_sons, aes(x = scaled_s, y = scaled_f)) +
  geom_point() +
  geom_smooth(method = "lm", color = "dodgerblue") +
  labs(x = "Son Height", y = "Father Height",
       title="Standardized Height Regressed with Perfect Correlation in Red") +
  theme_bw()+
  geom_abline(slope=1, col='red')
```

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




```
  #annotate(geom="text", x=65, y=75, label=paste("R^2== ",round(summary(lm_fit_f)$r.squared, digits=2))
  theme(text= element_text(size=14))
```

```
## List of 1
##  $ text:List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : num 14
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi FALSE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```

With standardized values, the correlation coefficient is identical regardless of which axis is chosen. The value less than the perfect correlation of 1 shows that we should always expect "reversion to mediocrity," even with more strongly correlated data.