**MA 354: Data Analysis I – Fall 2021**
**Homework 4:**

*Complete the following opportunities to use what we've talked about in class. These questions will be graded for correctness, communication and succinctness. Ensure you show your work and explain your logic in a legible and refined submission.*

0. **Complete weekly diagnostics.**

1. On its website, Ozempic, a medication for lowering the risk of major cardiovascular events (e.g., heart attack, stroke, etc.), states that

   - 66% of people taking 0.5 mg Ozempic
   - 73% of people taking 1 mg Ozempic
   - 40% of people taking 100 mg Januvia

   reached an A1C under 7%, noting higher A1C is indicative of higher risk of heart disease.

   (a) Explain why this statement alone isn't enough to conclude whether there is a statistically significant difference among the treatments.

```
"To assess any significant difference we need ot run a t-sample proportion test, but since we only

## [1] "To assess any significant difference we need ot run a t-sample proportion test, but since
```

   (b) The statement on Ozempic's website comes from a phase 3a randomized double-blind study. Ahrén et al. (2017) reports that 409 received Ozempic (0.5 mg), 409 received Ozempic (1 mg), and 407 received Januvia (100 mg).

      i. Determine whether there is sufficient evidence of a difference in rates of attaining an A1C under 7% across treatments.

```
#https://pubmed.ncbi.nlm.nih.gov/28385659/
#Mean baseline HbA1c was 8·1%
#2)

### Remember to write about the assumptions of doing t-sample prop.test#####
# Successful and total cases for 0.5 mg Ozempic
x1 = round(0.66*409)
n1 = 409

# Successful and total cases for 1 mg Ozempic
x2 = round(0.73*409)
n2 = 409

# Successful and total cases for 100 mg Januvia
x3 = round(0.4*407)
n3 = 407

prop.test(x = c(x1, x2, x3), n = c(n1, n2, n3))
##
##  3-sample test for equality of proportions without continuity
##  correction
##
## data:  c(x1, x2, x3) out of c(n1, n2, n3)
## X-squared = 102.7, df = 2, p-value < 2.2e-16
## alternative hypothesis: two.sided
## sample estimates:
##    prop 1    prop 2    prop 3
## 0.6601467 0.7310513 0.4004914
```

```
# There is a significant p-value to support a difference ()


#t sample proportion test
#DO IT
```

ii. Perform a follow-up analysis for comparing treatments. If you were at high risk for cardiovascular events, which medication would you want to take.

```
pairwise.prop.test(x = c(x1, x2, x3), n = c(n1, n2, n3))
```

```
##
##  Pairwise comparisons using Pairwise comparison of proportions
##
## data:  c(x1, x2, x3) out of c(n1, n2, n3)
##
##   1       2
## 2 0.033   -
## 3 3.6e-13 < 2e-16
##
## P value adjustment method: holm
```

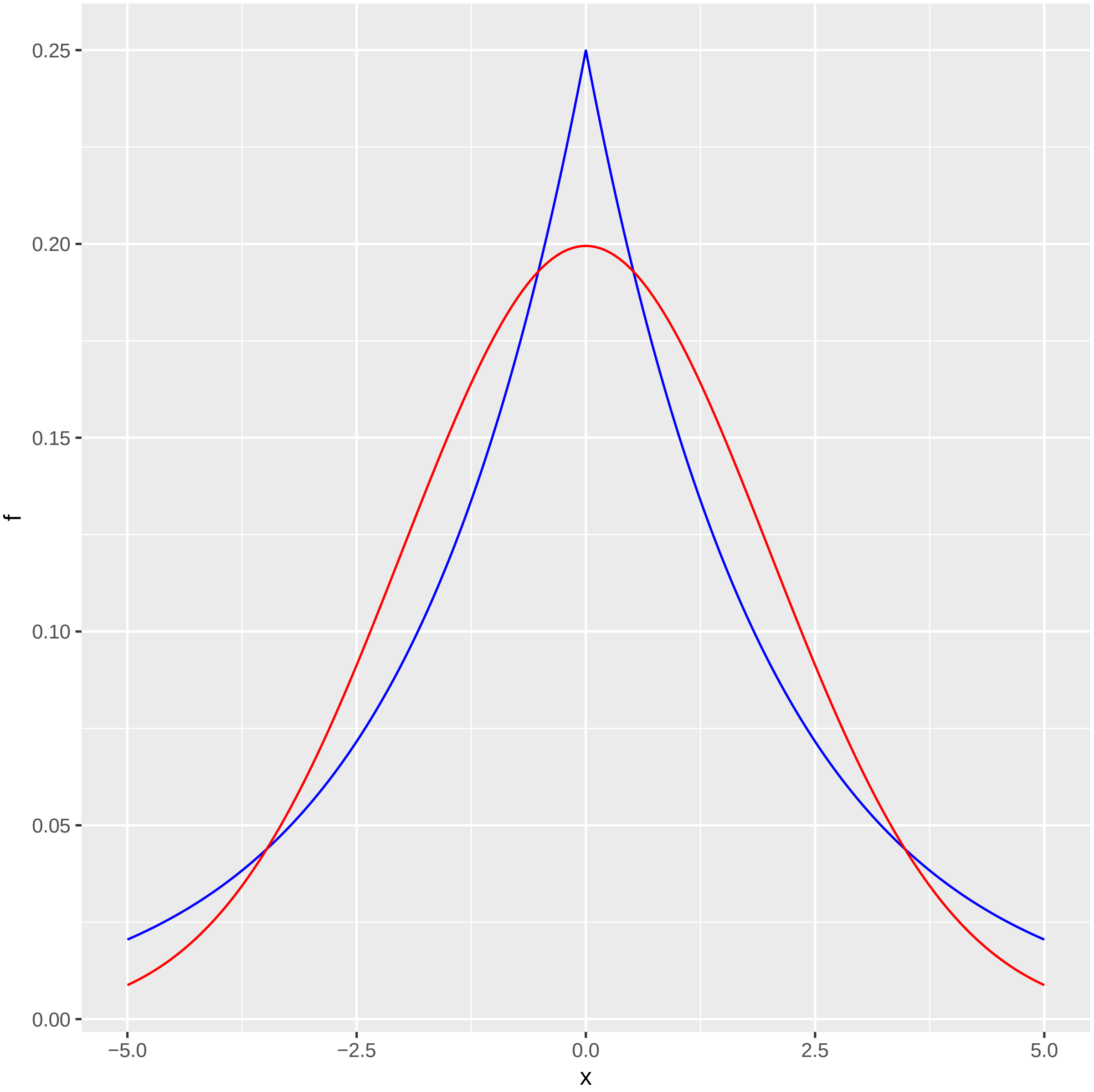
```
# There is a significant difference between 2 and 3 and 1 and 3 success proportions. So
# 1 and 2 are better than 3. But if we have alpha = 0.05 then there is also a significant
# difference between 1 and 2, in which case 2 (1 mg of Ozempic) is better.
```

2. Is the ANOVA really robust to Normality? Equal sample size? Equal variance? To assess this we'll check the ability of ANOVA to detect differences in a sample and retain the $\alpha = 0.05$ across different settings. This homework question was motivated by Blanca et al. (2017) who published a simulation study about ANOVA.

   **Remark:** My professor in graduate school always told me that I didn't have to memorize any results, I could just derive them. The data analysis analog to this is that if we have any questions about how a model works under a given condition (or broken assumption) we can just simulate it!

   (a) Plot the Laplace distribution with $m = 0$ and $s = 2$; the PDF of this distribution is cataloged in `R` as `dlaplace()` in the rmutil package, which you'll need to install and load. Superimpose the graph of the Gaussian distribution with $\mu = 0$ and $\sigma = 2$. Comment on the differences you see and what you think might happen if the data are Laplace distributed instead of the Gaussian distribution.

```r
library(rmutil)
library(tidyverse)

ggdat <- data.frame(x=seq(-5, 5, length.out=5000))%>%
mutate(f=dlaplace(x, m=0, s=2),
       f1=dnorm(x, mean=0, sd=2))

#set the legend
ggplot(ggdat, aes(x=x, y=f))+
geom_line(color="blue")+
geom_line(aes(y=f1), color="red")
```



(b) Conduct a simulation study using the Laplace distribution. To do so, complete the following 1000 times and report the proportion of times the data lead to a rejection of the null hypothesis.

The most efficient way to complete this question (including the other parts) is to write a function that completes the following.

- **Input:**
  - `rand.n=FALSE` – a logical object denoting whether the sample size is random or not. False by default. See part (f).
  - `rand.s=FALSE` – a logical object denoting whether the dispersion equal or not not. False by default. See part (g).
  - `equal.m=TRUE` – a logical object denoting whether the location parameters should be equal (part b) or different (part c). TRUE by default.
  - `n=5` – the desired sample size if not random. Five by default.
- **Loop the following tasks 1000 times:**
  - Generate $t = 4$ samples of size $n$ drawn independently from the laplace distribution with $m$ and $s$ which can be done using `rlaplace()` function from the rmutil package (Swihart and Lindsey, 2020). Specify $n$, $m$, and $s$ based on the values of the logical variables described above.

    - Perform the ANOVA procedure on these generated data.
    - Store whether the test resulted in a rejected null hypothesis or not.
- **Return:**
    - Your function should return the proportion of the 1000 ANOVA tests that resulted in a rejected null hypothesis.

Comment on the results of this simulation completed in the default case where $m_1 = m_2 = m_3 = m_4 = 0$, $s_1 = s_2 = s_3 = s_4 = 2$, and $n_1 = n_2 = n_3 = n_4 = 5$

```r
library(rstatix)
aovFunc <- function(rand.n=FALSE, rand.s=FALSE, equal.m=TRUE, n=5, loop=0,
                    welch=FALSE){
  alpha<-0.05
  count=0
  if(rand.n){
    n1=round(runif(1, min=5, max=100),0)
    n2=round(runif(1, min=5, max=100),0)
    n3=round(runif(1, min=5, max=100),0)
    n4=round(runif(1, min=5, max=100),0)
  }
  else{
    n1=n
    n2=n
    n3=n
    n4=n
  }

  if(rand.s){
    #DOUBLE CHECK WITH PROFESSOR
    s1=rgamma(1, 2, 1)
    s2=rgamma(1, 2, 1)
    s3=rgamma(1, 2, 1)
    s4=rgamma(1, 2, 1)
  }else{
    s1=1
    s2=1
    s3=1
    s4=1
  }
  for(i in 1:loop){
    if(equal.m){
      t1<-rlaplace(n=n1, m=0, s=s1)
      label1<-"T1"

      t2<-rlaplace(n=n2, m=0, s=s2)
      label2<-"T2"

      t3<-rlaplace(n=n3, m=0, s=s3)
      label3<-"T3"

      t4<-rlaplace(n=n4, m=0, s=s4)
      label4<-"T4"
    }

    else{
      t1<-rlaplace(n=n1, m=0, s=s1)
      label1<-"T1"
      t2<-rlaplace(n=n2, m=0, s=s2)
      label2<-"T2"

      t3<-rlaplace(n=n3, m=0, s=s3)
      label3<-"T3"

      t4<-rlaplace(n=n4, m=1, s=s4)
      label4<-"T4"
    }

    dat<-data.frame(value=c(t1, t2, t3, t4),
                    group=c(rep(c("T1", "T2", "T3", "T4"),
                                                         times=c(length(t1),
                                                                 length(t2),
                                                                 length(t3),
                                                                 length(t4)))))

    if(!welch){
      anova<-summary(aov(value~group, data=dat))
      sum_test <- unlist((anova))
      p.value<-sum_test["Pr(>F)1"]
      #print(p.value) #bugtest
    }else{
      anova_w<-welch_anova_test(value~group, data=dat)
      p.value<-anova_w$p
    }

    if(p.value<0.05){
     count=count+1
    }
  }
  count/loop
}
aovFunc(loop=100)
```

```
## [1] 0.03
```

(c) Repeat the simulation study in (b-d), except with different means; i.e., $m_1 = m_2 = m_3 = 0$, and $m_4 = 1$. Comment on the results of this simulation.

```
aovFunc(equal.m=FALSE, loop=100)
```

```
## [1] 0.22
```

(d) Repeat the simulation study in (b-c), except with $n = 15$. Comment on the results of this simulation.

```
aovFunc(equal.m=TRUE, loop=100, n=15)
```

```
## [1] 0.07
```

```
aovFunc(equal.m=FALSE, loop=100, n=15)
```

```
## [1] 0.46
```

(e) Repeat the simulation study in (b-c), except with $n = 50$. Comment on the results of this simulation.

```
aovFunc(equal.m=TRUE, loop=100, n=50)
```

```
## [1] 0.04
```

```
aovFunc(equal.m=FALSE, loop=100, n=50)

## [1] 1
```

(f) Repeat (b-e), except randomly select the sample size for each group by selecting $n$ from the uniform(5,100) distribution. This will help us assess the robustness of the equal sample size assumption in the Laplace population distribution case. Comment on the results of this simulation.

```
aovFunc(equal.m=TRUE, rand.n=TRUE, loop=100)

## [1] 0.07

aovFunc(equal.m=FALSE, rand.n=TRUE, loop=100)

## [1] 0.98
```

(g) Repeat (b-f), except randomly select the dispersion for each group by selecting $s$ from the gamma(2,1) distribution. This will help us assess the robustness of the equal variance assumption in the Laplace population distribution case. Comment on the results of this simulation.

```
aovFunc(equal.m=TRUE, rand.n=TRUE, rand.s=TRUE, loop=100)

## [1] 0.02

aovFunc(equal.m=FALSE, rand.n=TRUE, rand.s=TRUE, loop=100)

## [1] 0.61
```

(h) Write a loop that conducts this simulation when (1) $s$ is fixed and (2) when $s$ is random, for the case where the means are unequal. The loop should be with respect to $n$, and should run for $n = 5$ to $n = 200$.

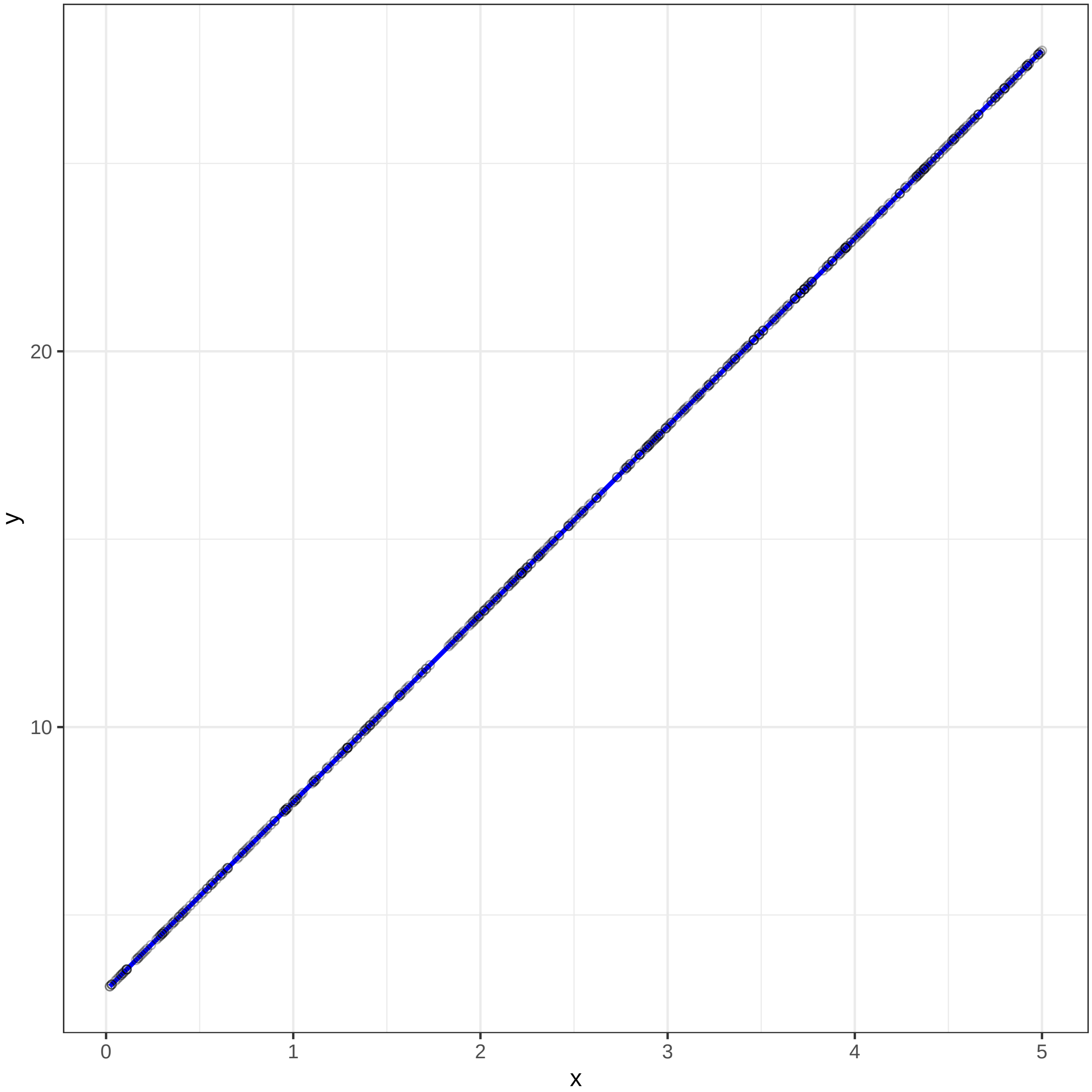
```
# results<-c()
# n<-10:50
# y1<-c()
# y2<-c()
# y3<-c()
#
# for(i in 10:50){
#   print(i)
#   unequal<-aovFunc(equal.m=FALSE, rand.s=TRUE, loop=100, n=i)
#   y1<-c(y1, unequal)
#   equal<-aovFunc(equal.m=FALSE, rand.s=FALSE, loop=100, n=i)
#   y2<-c(y2, equal)
#
#   wUnequal<-aovFunc(equal.m=FALSE, rand.s=TRUE, loop=100, n=i, welch=T)
#   y3<-c(y3, wUnequal)
# }
#
# ggdat<-data.frame(x=n, equal.s=y2, unequal.s=y1,
#                   welch=y3)
# ggplot(ggdat, aes(x=x))+
#   geom_line(aes(y=equal.s), color="red")+
#   geom_line(aes(y=unequal.s), color="blue")+
#   geom_line(aes(y=welch), color="black")+
#   labs(title="Everybody hates Welch")
```

**Note:** This can take some computation time, you'll want to run it and save the image as a .pdf so you can load it instead of rerunning the code.

3. Complete the following parts. This will lead you through the simulation of data, fitting regression lines and evaluating the assumptions.

   (a) Fit a model to the following simulated data. Make observations about the model equation and the Pearson correlation.

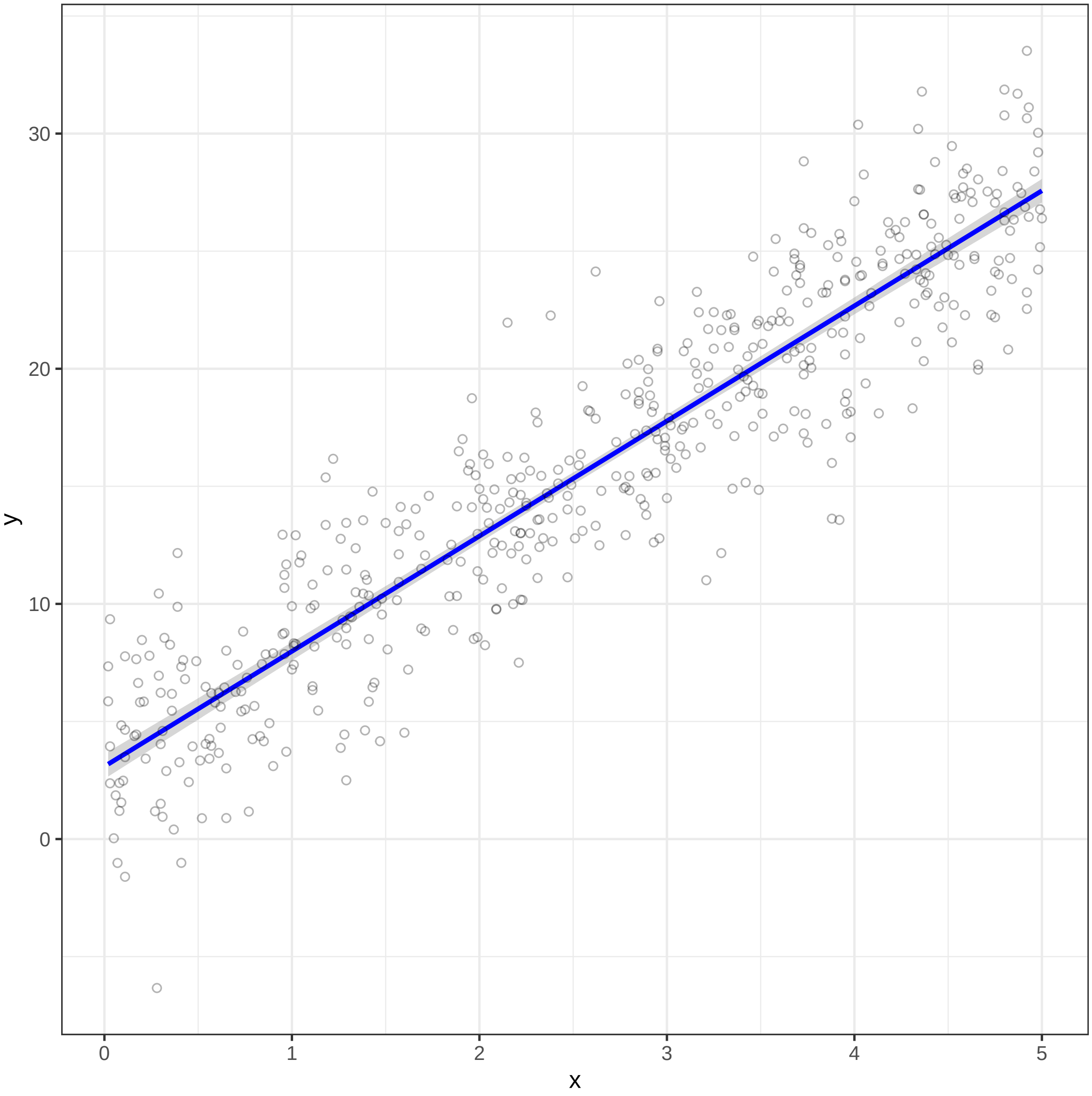
```
n=500
x<-sample(x = seq(0,5,0.01), size=n, replace=T)
y<-5*x + 3

ggdat<-data.frame(x=x, y=y)
ggplot(ggdat, aes(x=x, y=y))+
  geom_smooth(color="blue",
              method="lm",
              formula=y~x)+
  geom_point(shape=1,
             alpha=.3)+
  theme_bw()
```



```
  #put pearson on the plot
```

(b) Fit a model to the following simulated data, now with added Normal error. Make observations about the model equation and the Pearson correlation in relation to (a).

```
e<-rnorm(n=n,mean=0,sd=3)
y2<-5*x + 3 + e

ggdat<-data.frame(x=x, y=y2)
ggplot(ggdat, aes(x=x, y=y))+
  geom_smooth(color="blue",
              method="lm",
              formula=y~x)+
  geom_point(shape=1,
             alpha=.3)+
  theme_bw()
```

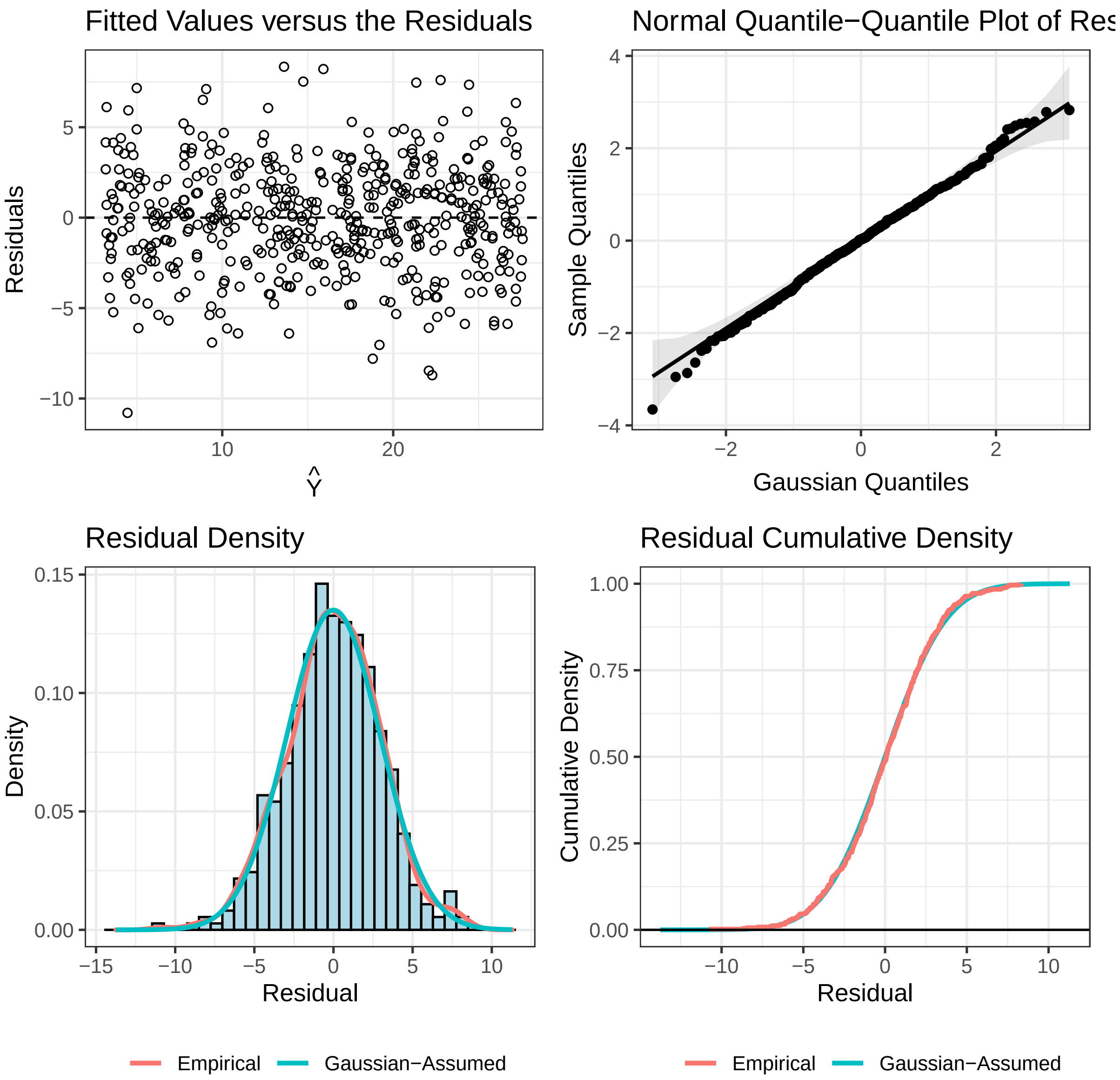


```
  #put pearson on the plot
#cor(x, y2, method="pearson")
```

(c) In the model of part (b), evaluate the normality and homogeneity of error terms. Note that we know both of these items to be true since we've taken $\epsilon \sim \mathrm{N}(\mu = 0, \sigma = 3)$.

```
#Preparing the functions!
library(patchwork)
source("https://cipolli.com/students/code/plotResiduals.R")
source("https://cipolli.com/students/code/plotInfluence.R")
```

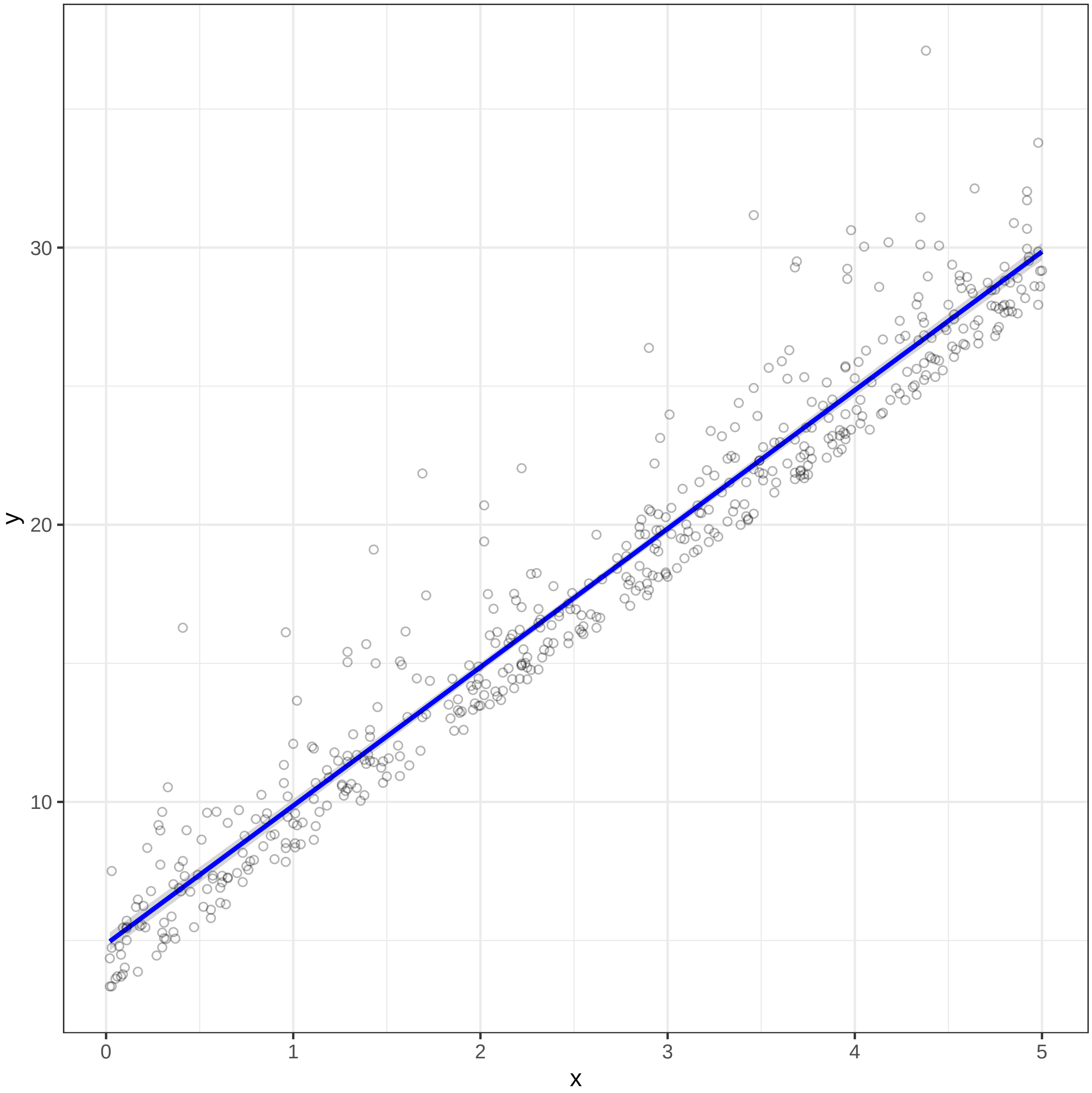
```
model1 <- lm(y2~x)
plotResiduals(model1)
```



```
#Variance is the same
#https://uc-r.github.io/assumptions_homogeneity#visualization
```

(d) Fit a model to the following simulated data, now with added exponential error. Make observations about the model equation and the Pearson correlation in relation to the model of part (b).

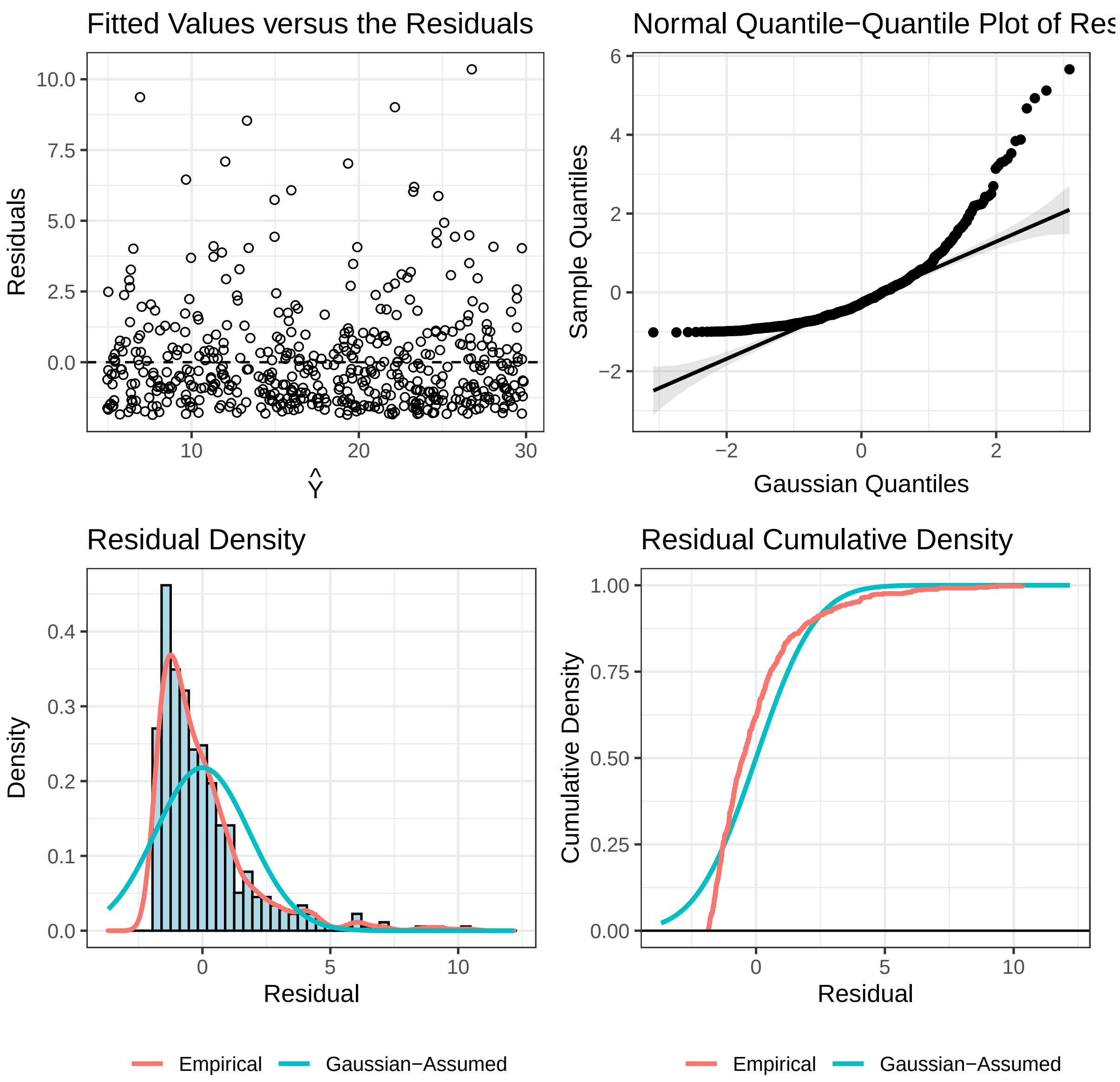
```r
e<-rexp(n=n,rate = 1/2)
y3<-5*x + 3 + e

ggdat<-data.frame(x=x, y=y3)
ggplot(ggdat, aes(x=x, y=y))+
  geom_smooth(color="blue",
              method="lm",
              formula=y~x)+
  geom_point(shape=1,
             alpha=.3)+
  theme_bw()
```



```r
  #put pearson on the plot
cor(x, y3, method="pearson")

## [1] 0.9701446
```

(e) In the model of part (d), evaluate the normality and homogeneity of error terms. Note that we know that common variance is true but we've taken $\epsilon \sim \exp(\beta = 2)$.

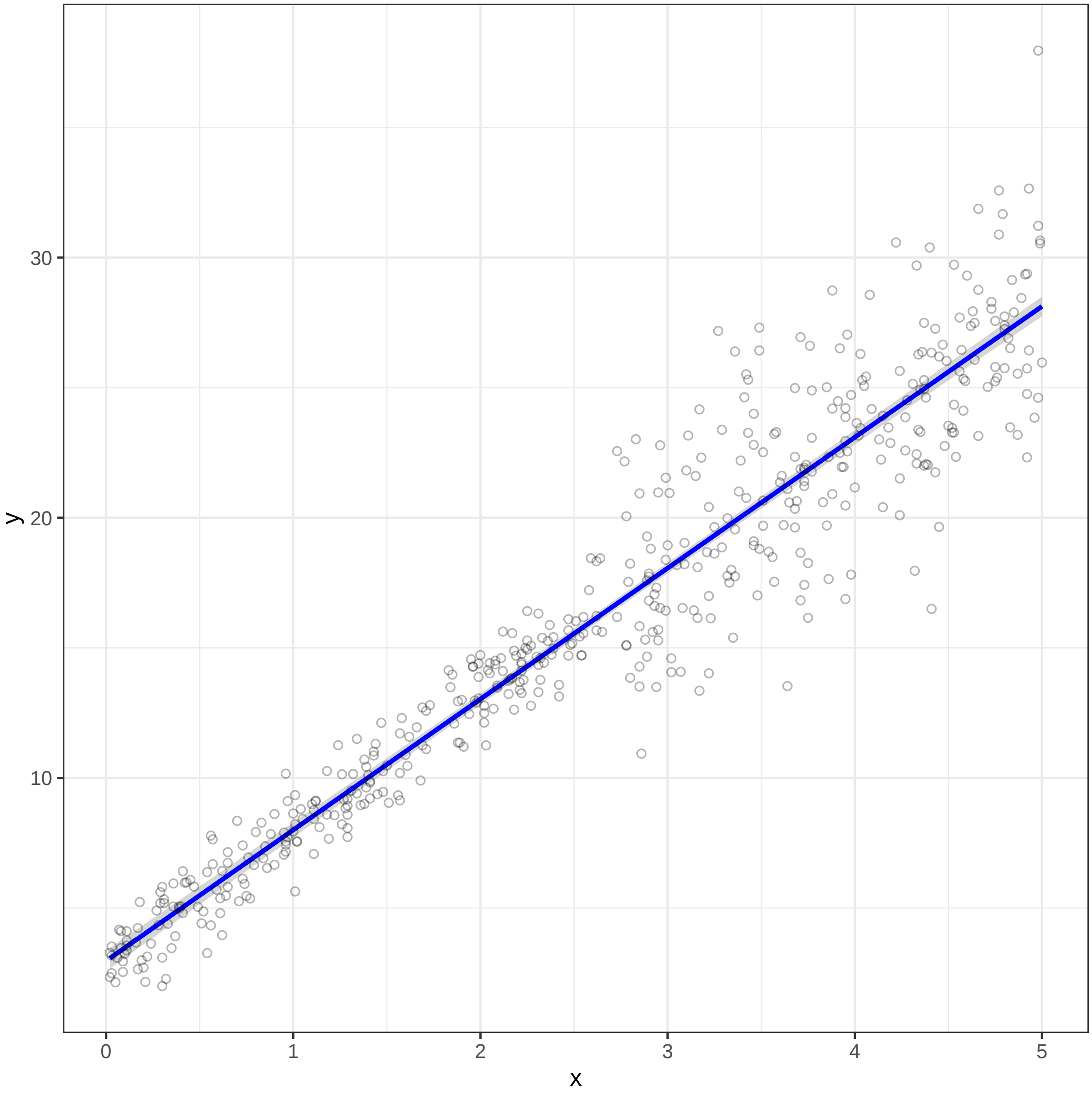
```
model2 <- lm(y3~x)
plotResiduals(model2)
```



```
#It's a skewed normal dist, but it's homogeneous.
```

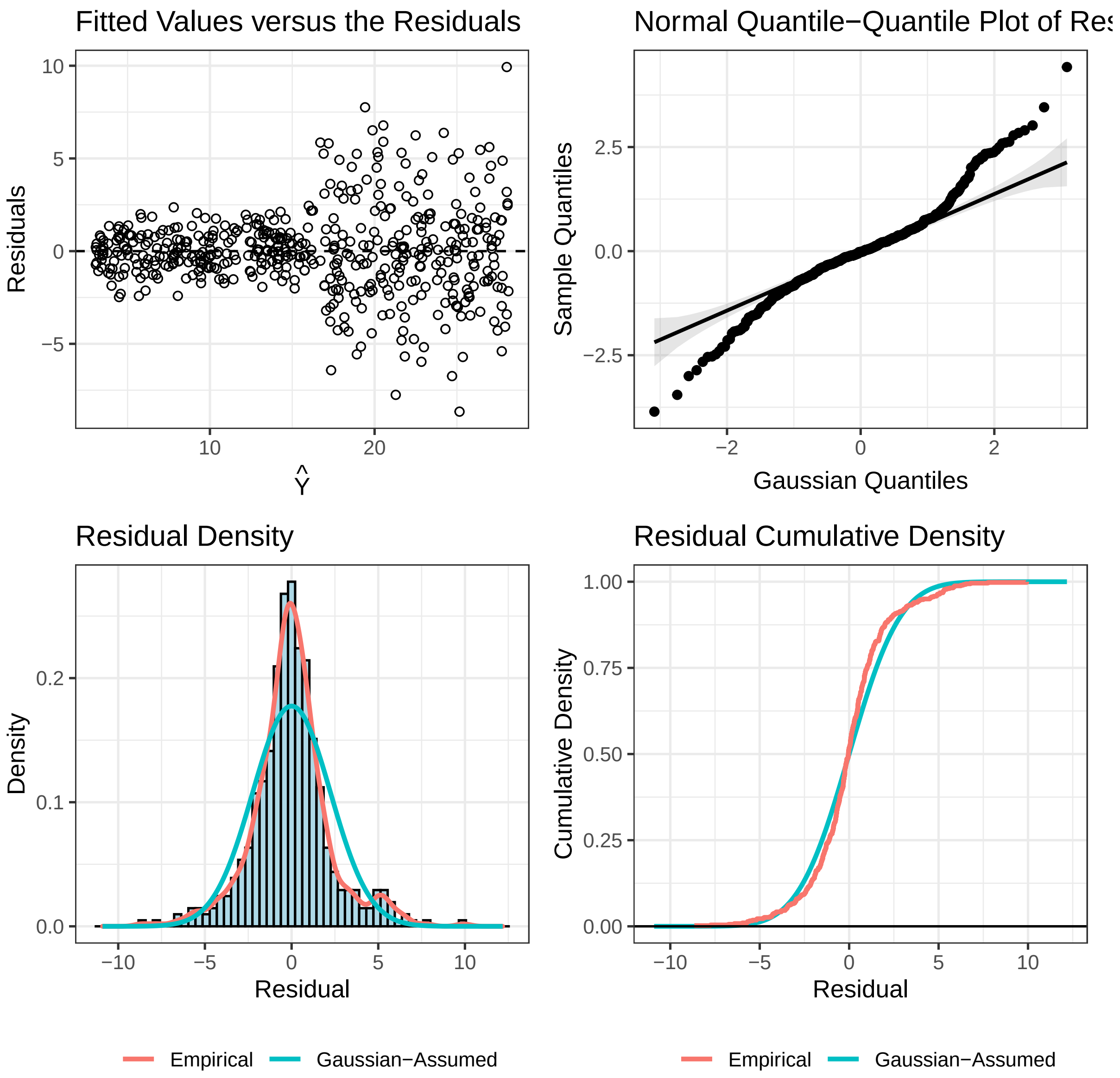
(f) Fit a model to the following simulated data, now with added Heteroskedastic normal error. Make observations about the model equation and the Pearson correlation in relation to the model of part (b).

```
x4<-x[order(x)]
e<-rnorm(n=n,mean=0,sd=c(rep(1,n/2),rep(3,n/2)))
y4<-5*x4 + 3 + e

ggdat<-data.frame(x=x4, y=y4)
ggplot(ggdat, aes(x=x, y=y))+
  geom_smooth(color="blue",
              method="lm",
              formula=y~x)+
  geom_point(shape=1,
             alpha=.3)+
  theme_bw()
```



(g) In the model of part (f), evaluate the normality and homogeneity of error terms. Note that we know that normality of error terms is true, but $\epsilon \sim \mathrm{N}(\mu = 0, \sigma = 1)$ for $x < \widehat{m}$ and $\epsilon \sim \mathrm{N}(\mu = 0, \sigma = 3)$ for $x > \widehat{m}$.

```
model3 <- lm(y4~x4)
plotResiduals(model3)
```



4. Consider the following simulation.

   (a) Plot the data simulated below. Assess the linear relationship.

```
library(tidyverse)
set.seed(7272)
n<-50
ggdat <- data.frame(x=sample(x=seq(0,100,0.01),size=n,replace=TRUE)) %>%
  mutate(y=3.5+2.1*x+rnorm(n=n,mean=0,sd=5))
```

   (b) Write out the population model.

```
#Y=B{0}+B{1}{x} + error
```

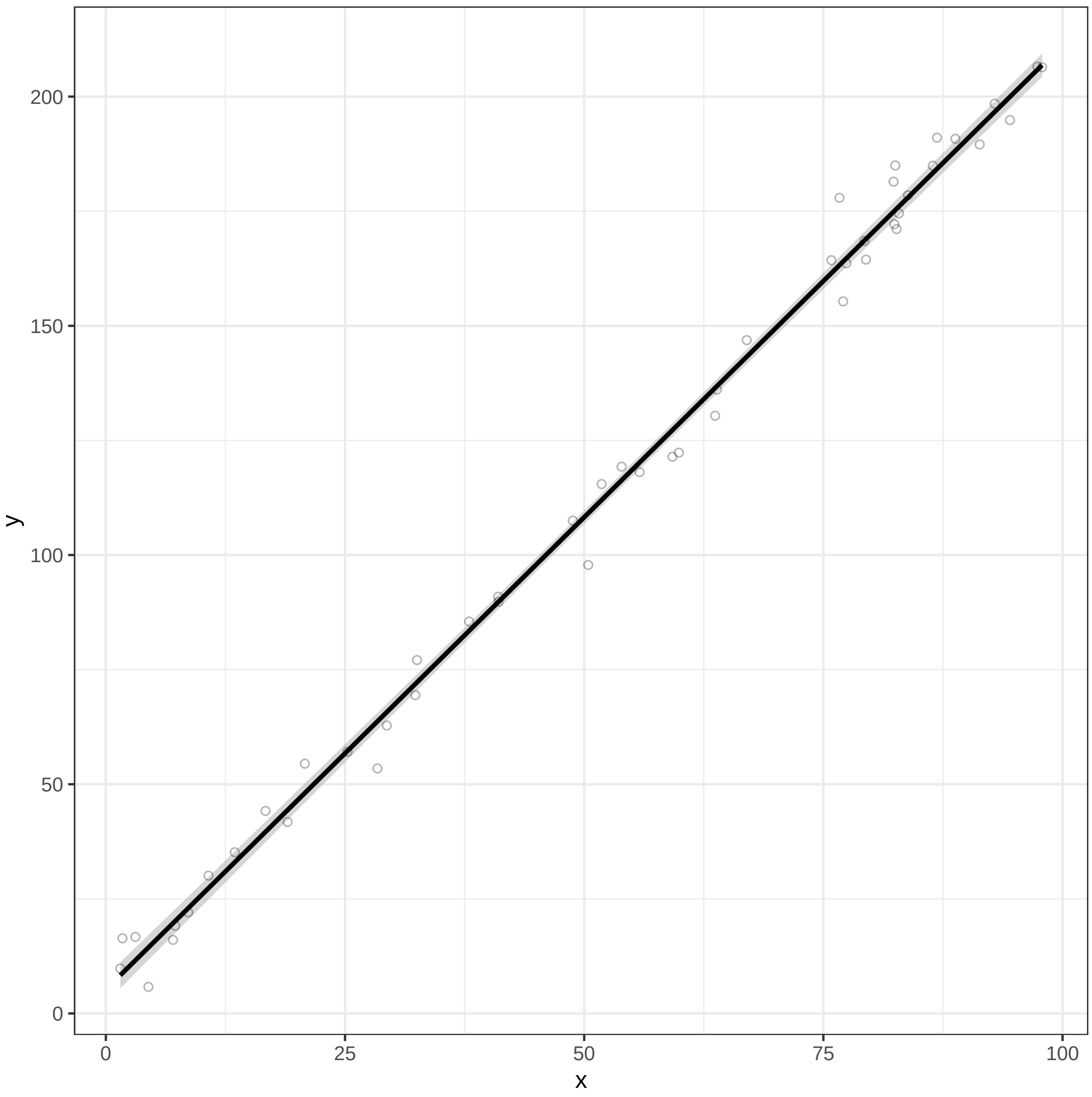
   (c) Fit the model based on the sample data and write out the sample model below.

```
#hat(Y)=hat(B{0})+hat(B){1}{x}
four.model<-lm(y~x, data=ggdat)
```

```
summary(four.model)

## 
## Call:
## lm(formula = y ~ x, data = ggdat)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -11.2855  -2.9153  -0.0545   2.7938  14.7084 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  5.22286    1.44261    3.62 0.000707 ***
## x            2.06056    0.02345   87.89  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 5.152 on 48 degrees of freedom
## Multiple R-squared:  0.9938,Adjusted R-squared:  0.9937 
## F-statistic:  7724 on 1 and 48 DF,  p-value: < 2.2e-16

#hat(B){0}=5.22
#hat(B){1}{x}=2.06

#hat(Y)=-5.22+2.06x

#Prediction = 5.22+2.06(x) + (random error)
```

(d) Add the regression line to the plot in black.

```
ggplot(ggdat, aes(x=x, y=y))+
  geom_smooth(color="black",
              method="lm",
              formula=y~x)+
  geom_point(shape=1,
             alpha=.3)+
  theme_bw()
```



(e) Interpret the $R^2$ of the model.

```
#Adjusted R-squared:  0.9937
#99% of the variance can be explained by the model we built.
```

(f) Interpret the overall $F$ test of the model.

```
#Look at p-value
```

(g) Interpret the coefficients of the model; are they what you would expect?

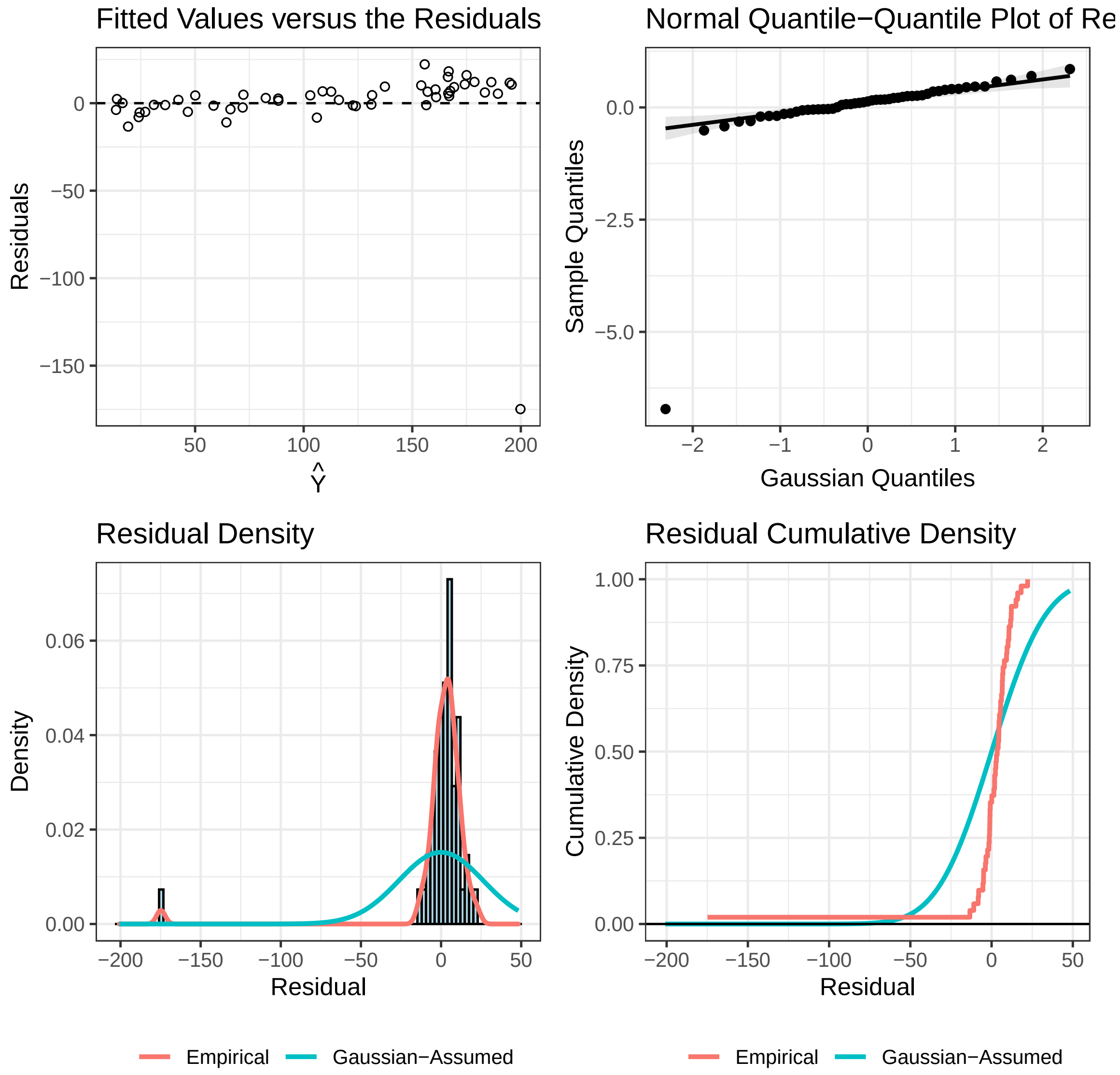
```
#Yeah, that's easy.
```

(h) Now, let's add a bad datapoint to the data.

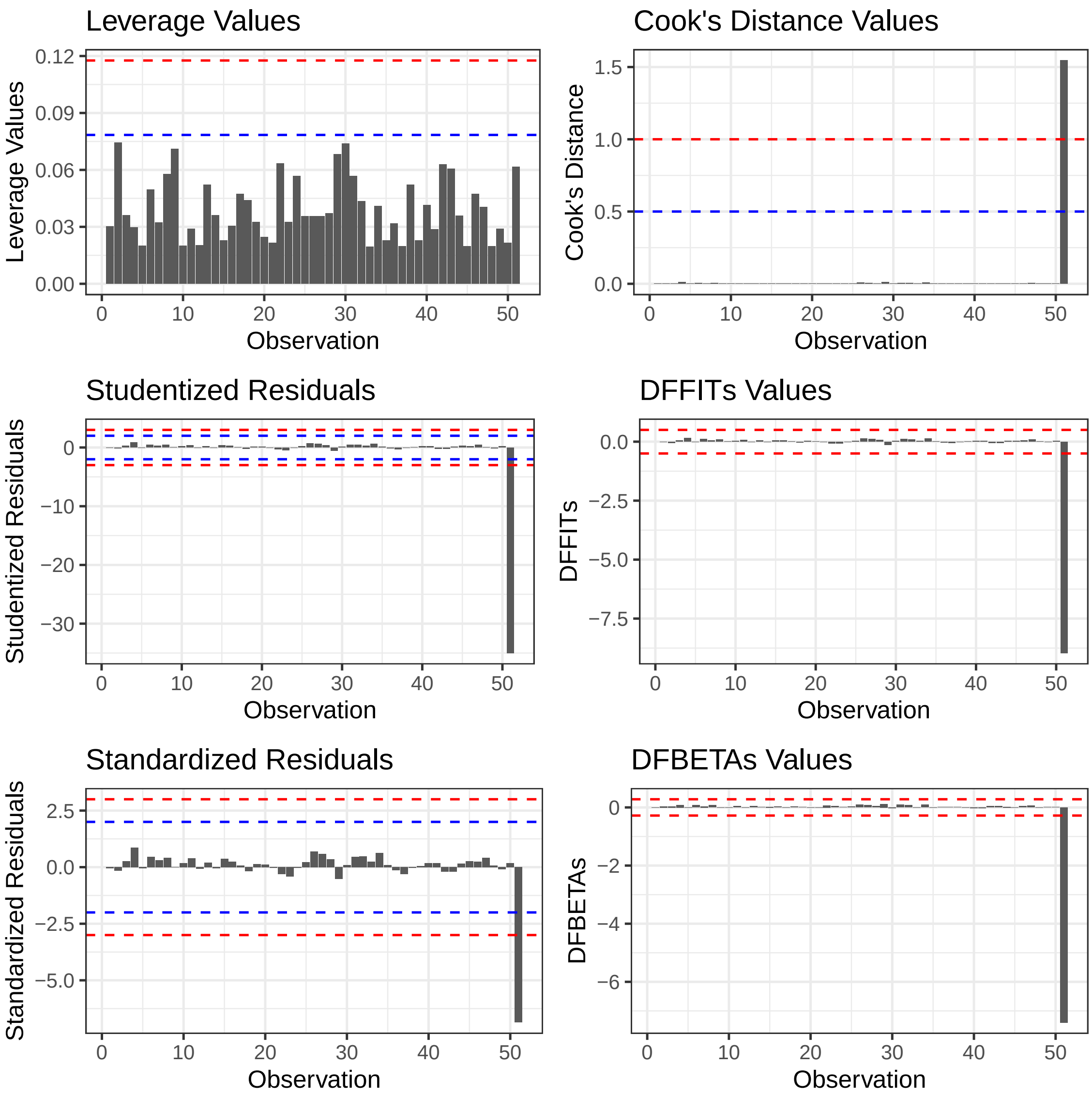
```
ggdat <- rbind(ggdat,      # original data
               c(100,25)) # bad observation
```

i. Briefly summarize how adding this data point affects parts (a)-(g).

```
#ggplot(ggdat, aes(x=x, y=y))+
#  geom_smooth(color="blue",
#              method="lm",
#              formula=y~x)+
#  geom_point(shape=1,
#             alpha=.3)+
#  theme_bw()
#there is one unusually small observation!
four.model<-lm(y~x, data=ggdat)
plotResiduals(four.model)
```

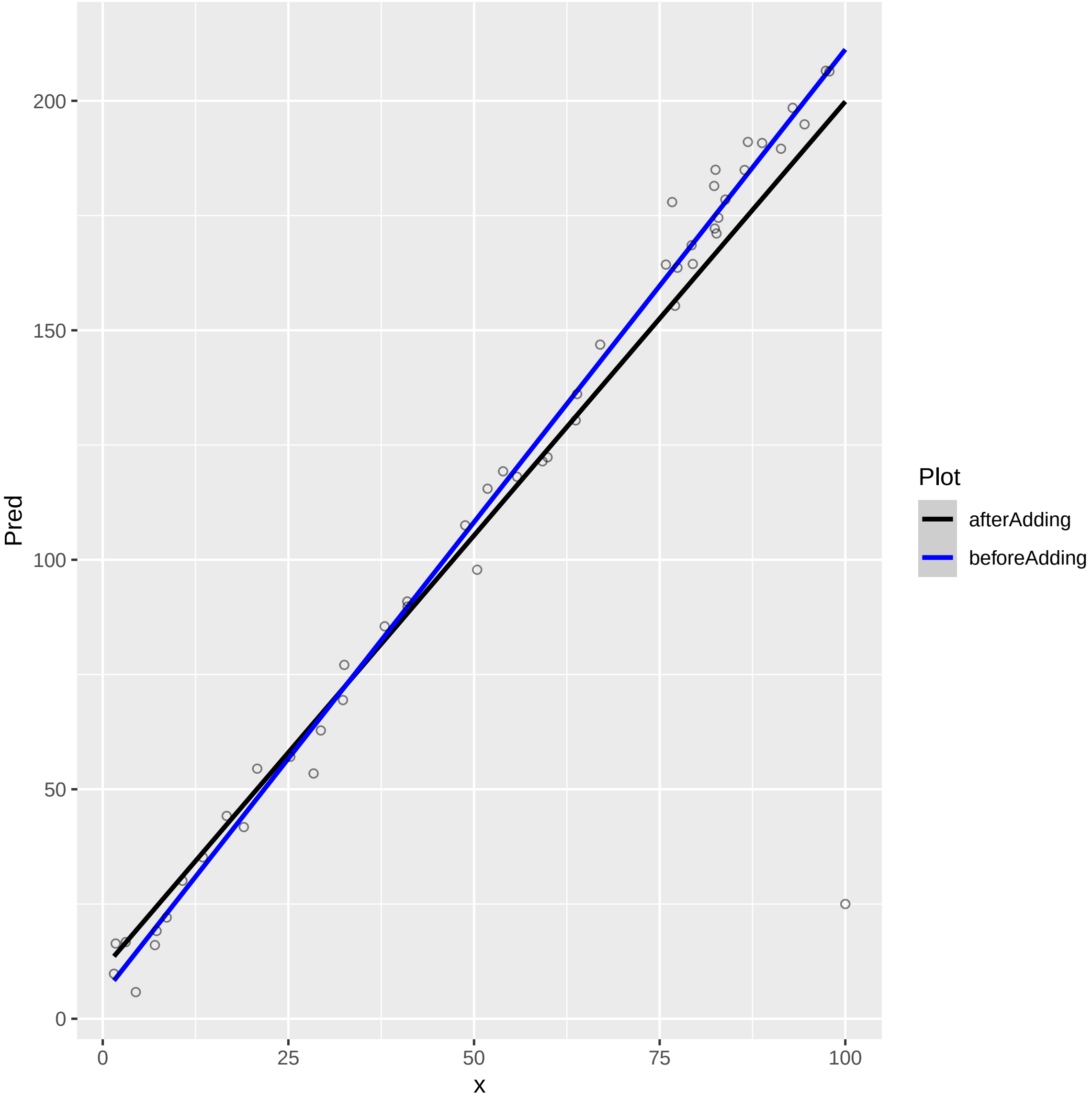


```
plotInfluence(four.model)
```



ii. Add the resulting regression line to the plot in part (d) in blue.

```
#5.2+(2.06*x)
#10.742+(1.891*x)

ggdat.final <- ggdat %>%
  mutate(beforeAdding = 5.2+(2.06*x),
     afterAdding = 10.742+(1.891*x))%>%
  pivot_longer(cols=ends_with("Adding"),
           names_to="Plot",
           values_to="Pred")


ggplot(ggdat.final, aes(x=x, y=Pred))+
  geom_smooth(aes(color=Plot),
          method="lm",
          formula=y~x)+
  geom_point(aes(y=y), shape=1,
         alpha=.3)+
  scale_color_manual(values=c("black", "blue"))
```



iii. Refit this model using several robust techniques for dealing with the bad observation. Create a plot that summarizes all the approaches taken, and use a metric to select the best model.

```
library(MASS)
library(sfsmisc)
#Hubert
#6.0391 + (2.0421 * x)
mod.hubert <- rlm(y ~ x, data=ggdat,
              psi=psi.huber)
summary(mod.hubert)
```

```
##
## Call: rlm(formula = y ~ x, data = ggdat, psi = psi.huber)
## Residuals:
##       Min        1Q    Median        3Q       Max
## -185.2474   -3.1410    0.5295    3.2745   15.3087
##
## Coefficients:
```

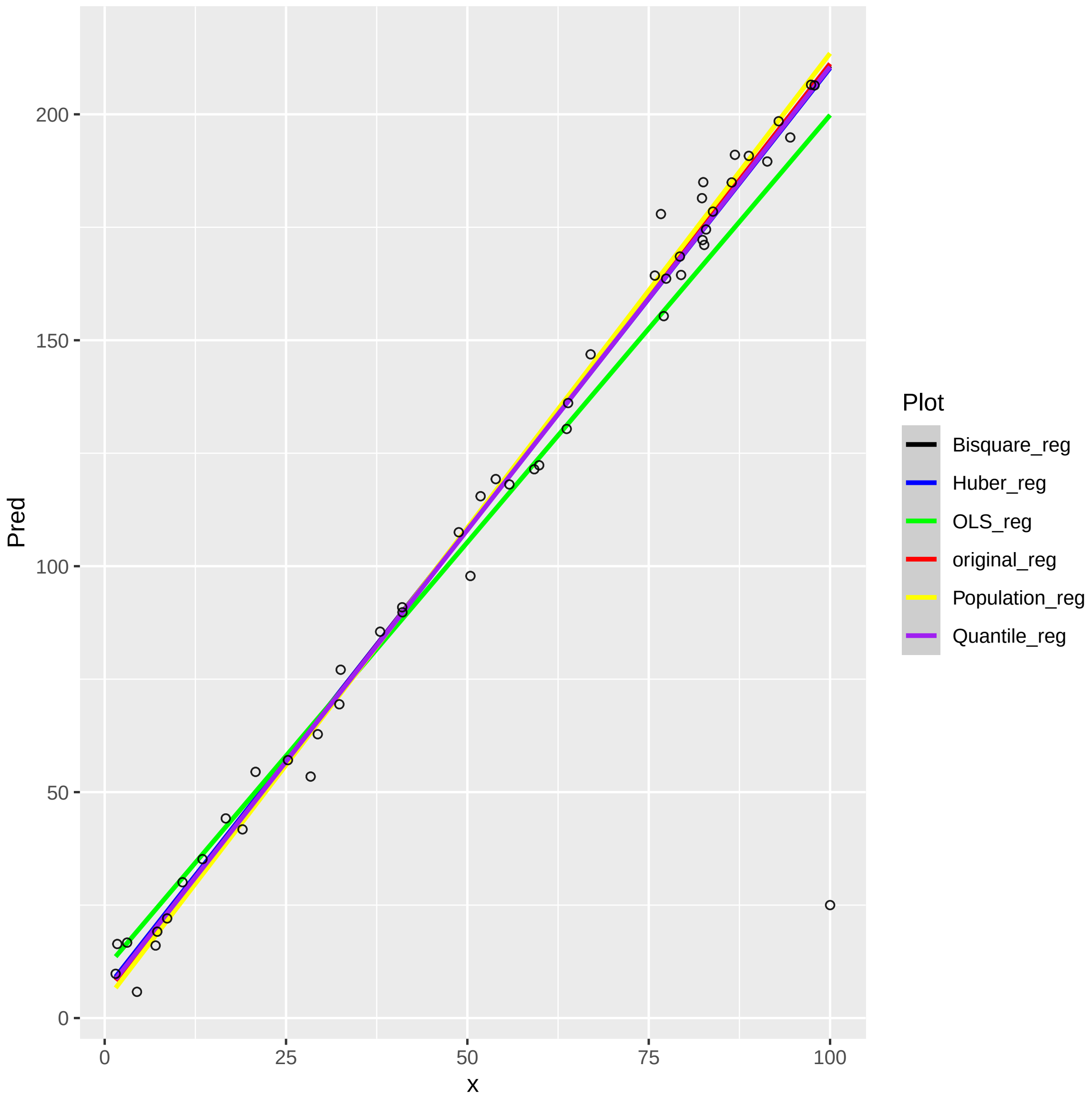
```
##              Value   Std. Error t value
## (Intercept)  6.0391  1.4224     4.2457
## x            2.0421  0.0228    89.7473
##
## Residual standard error: 4.747 on 49 degrees of freedom
f.robftest(mod.hubert, var="x")
##
##  robust F-test (as if non-random weights)
##
## data:  from rlm(formula = y ~ x, data = ggdat, psi = psi.huber)
## F = 7932.4, p-value < 2.2e-16
## alternative hypothesis: true x is not equal to 0
```

```
#5.7016 + (2.0504 * x)
mod.bisquare<-rlm(y ~ x, data=ggdat,
                  psi=psi.bisquare)

summary(mod.bisquare)
##
## Call: rlm(formula = y ~ x, data = ggdat, psi = psi.bisquare)
## Residuals:
##      Min       1Q   Median       3Q      Max
## -185.739   -3.265    0.056    3.033   15.011
##
## Coefficients:
##              Value   Std. Error t value
## (Intercept)  5.7016  1.4342     3.9755
## x            2.0504  0.0229    89.3722
##
## Residual standard error: 4.607 on 49 degrees of freedom
f.robftest(mod.bisquare, var="x")
##
##  robust F-test (as if non-random weights)
##
## data:  from rlm(formula = y ~ x, data = ggdat, psi = psi.bisquare)
## F = 7888.7, p-value < 2.2e-16
## alternative hypothesis: true x is not equal to 0
```

```
library(quantreg)
## Loading required package:  SparseM
##
## Attaching package:  'SparseM'
## The following object is masked from 'package:base':
##
##     backsolve
## Warning in .recacheSubclasses(def@className, def, env):  undefined subclass "numericVector"
of class "Mnumeric"; definition not updated
#5.52 + (2.05*x)
mod.quant <- rq(y~x, data=ggdat)
summary(mod.quant, se = "ker")
##
## Call: rq(formula = y ~ x, data = ggdat)
##
## tau: [1] 0.5
##
```

```
## Coefficients:
##              Value    Std. Error t value  Pr(>|t|)
## (Intercept)  5.52886  2.88708    1.91504  0.06133
## x            2.05271  0.04751   43.20164  0.00000
```

```
#6.0391 + (2.0421 * x) | Huber
#5.7016 + (2.0504 * x) | Bisquare
#5.52 + (2.05*x) | Quantile

ggdat.compare <- ggdat %>%
  mutate(original_reg = 5.2+(2.06*x),
         OLS_reg = 10.742+(1.891*x),
         Huber_reg = 6.0391 + (2.0421 * x),
         Bisquare_reg = 5.7016 + (2.0504 * x),
         Quantile_reg = 5.52 + (2.05*x),
         Population_reg = 3.5+(2.1*x))%>%
  pivot_longer(cols=ends_with("_reg"),
               names_to="Plot",
               values_to="Pred")


ggplot(ggdat.compare, aes(x=x, y=Pred))+
  geom_smooth(aes(color=Plot),
              method="lm",
              formula=y~x)+
  geom_point(aes(y=y), shape=1,
             alpha=.3)+
  scale_color_manual(values=c("black", "blue", "green", "red", "yellow", "purple"))
```



```
library(Metrics)

#rmse(ggdat£y, predict(mod.quant))
#rmse(ggdat£y, predict(mod.hubert))
#rmse(ggdat£y, predict(mod.bisquare))


#rmse(test1, test2)
```

```
#test1
#test2
#rmse(ggdat£y, predict(mod.hubert))
```

# References

Ahrén, B., Masmiquel, L., Kumar, H., Sargin, M., Karsbøl, J. D., Jacobsen, S. H., and Chow, F. (2017). Efficacy and safety of once-weekly semaglutide versus once-daily sitagliptin as an add-on to metformin, thiazolidinediones, or both, in patients with type 2 diabetes (sustain 2): a 56-week, double-blind, phase 3a, randomised trial. *The lancet Diabetes & endocrinology*, 5(5):341–354.

Blanca, M. J., Alarcón, R., Arnau, J., Bono, R., and Bendayan, R. (2017). Non-normal data: Is anova still a valid option? *Psicothema*, 29(4):552–557.

Swihart, B. and Lindsey, J. (2020). *rmutil: Utilities for Nonlinear Regression and Repeated Measurements Models*. R package version 1.1.5.