LeaRning about Statistics - YEAH!!!

Correlation, Scatterplots and Regression!

Zachary Thompson
Moffitt Cancer Center

July 01, 2022

What you will learn to run

- Scatterplots
- Correlation
- Univariate regresssion
- MultivarIABLE reression

Data prep: here()

```
here()
```

```
[1] "F:/myGitRepo/Intro_to_R_2022"
```

```
here("data")
```

```
[1] "F:/myGitRepo/Intro_to_R_2022/data"
```

```
here("data","tcga-clinical.txt")
```

```
[1] "F:/myGitRepo/Intro_to_R_2022/data/tcga-clinical.txt"
```

```
here("data","tcga-gene-exp.txt")
```

```
[1] "F:/myGitRepo/Intro_to_R_2022/data/tcga-gene-exp.txt"
```

Data prep: load

```
clinical <- read_csv(file = here("data","tcga_clinical.txt"))
geneexp <- read_csv(file = here("data","tcga_gene_exp.txt"))
```

Data prep: merge

```
tcgatest <- left_join(clinical, geneexp)
tcga <- left_join(clinical, geneexp, by = "bcr_patient_barcode")

   intersect(names(clinical), names(geneexp))
```

```
[1] "bcr_patient_barcode"
```

```
   dim(tcgatest)
```

```
[1] 1043   33
```

```
   dim(tcga)
```

```
[1] 1043   33
```

Data prep: mutate

```
  tcga <- tcga %>% mutate(
  smoking = case_when(
    tobacco_smoking_history %in% c(
      "Current reformed smoker for < or = 15 years",
      "Current reformed smoker for > 15 years",
      "Current Reformed Smoker, Duration Not Specified"
    ) ~ "Former",
    tobacco_smoking_history %in% c("Current smoker") ~ "Current",
    tobacco_smoking_history %in% c("Lifelong Non-smoker") ~ "Never",
    is.na(tobacco_smoking_history) ~ NA_character_
  )
)
```

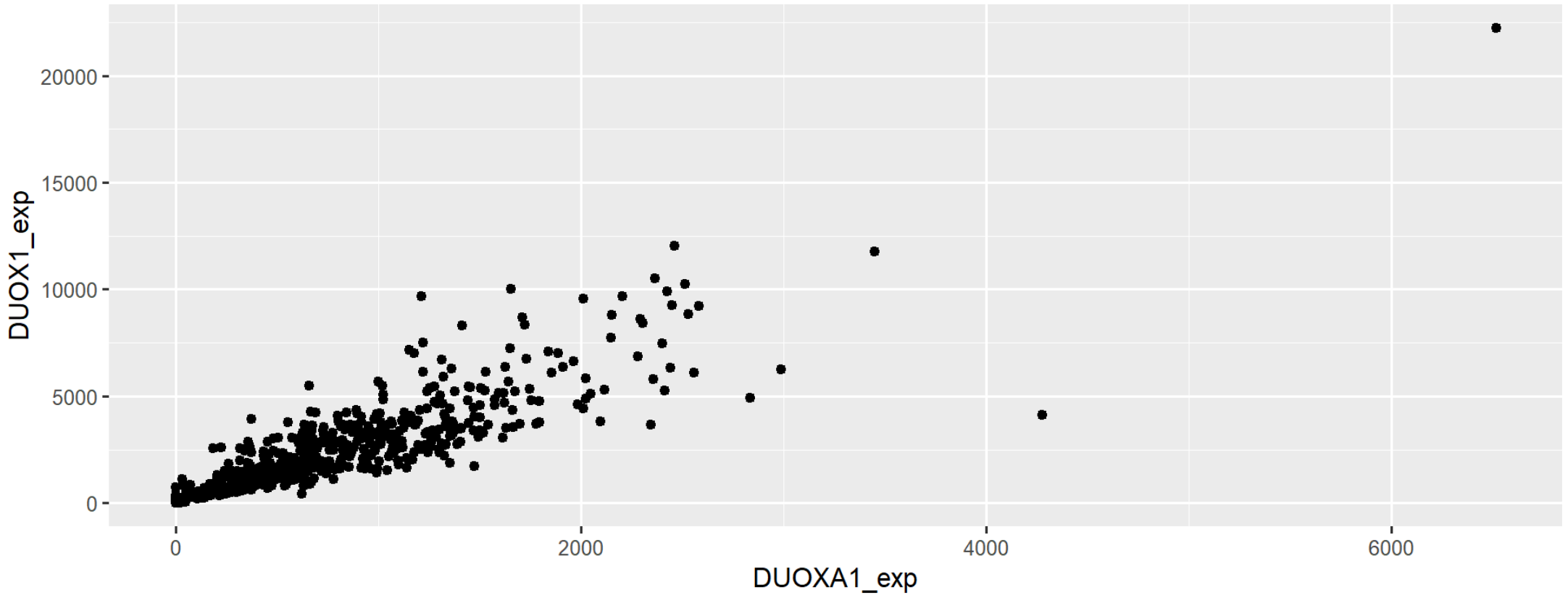
Scatter plots

R Code | Plot

```
ggplot(tcga) +
  aes(DUOXA1_exp, DUOX1_exp) +
  geom_point()
```

Scatter plots

R Code | Plot



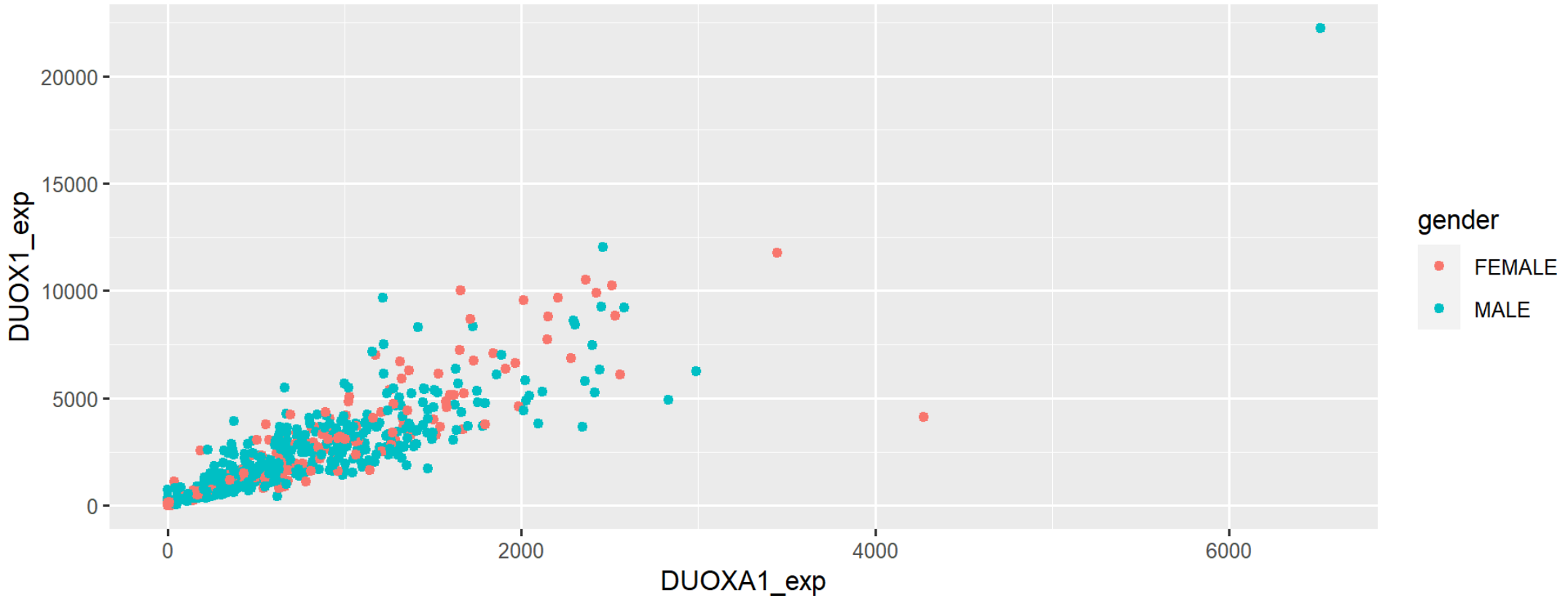
Scatter plots

R Code | Plot

```
ggplot(tcga) +
  aes(DUOXA1_exp, DUOX1_exp) +
  geom_point(aes(colour = gender))
```

Scatter plots

R Code | Plot



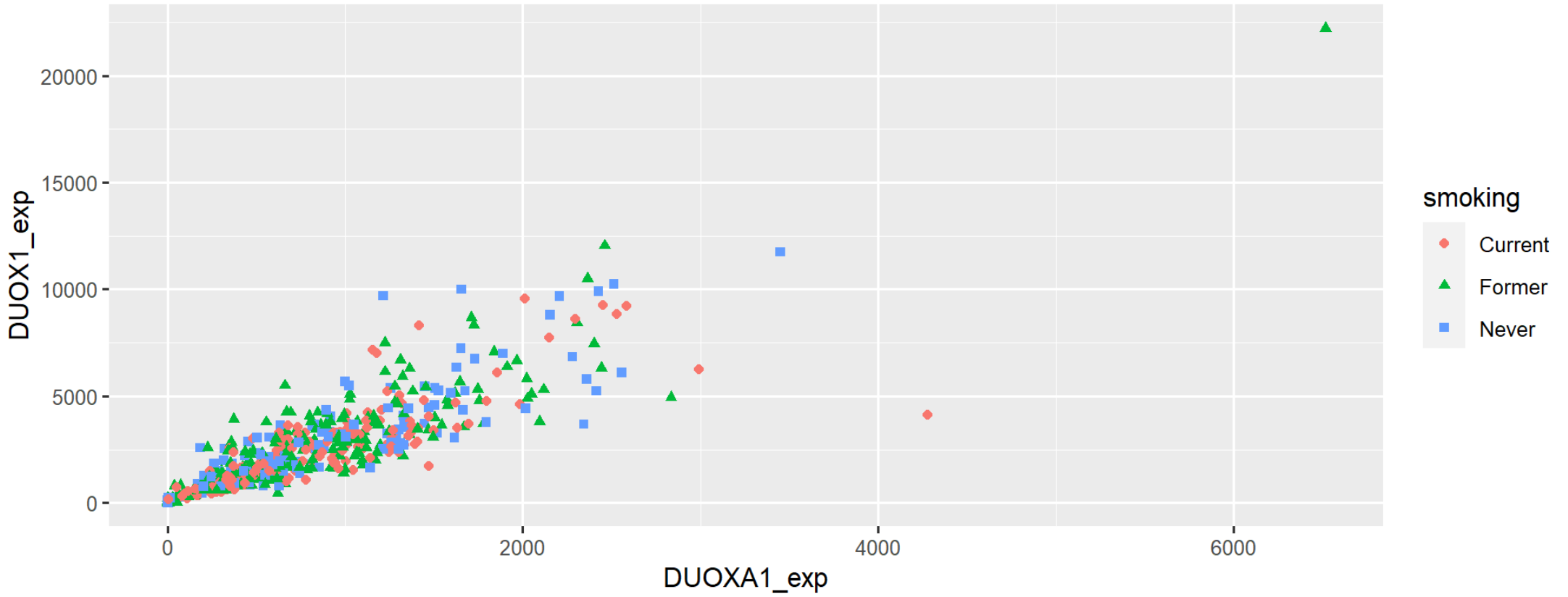
Scatter plots

R Code | Plot

```
ggplot(tcga %>% filter(!is.na(smoking))) +
  aes(DUOXA1_exp, DUOX1_exp) +
  geom_point(aes(shape = smoking, colour = smoking))
```

Scatter plots

R Code | Plot



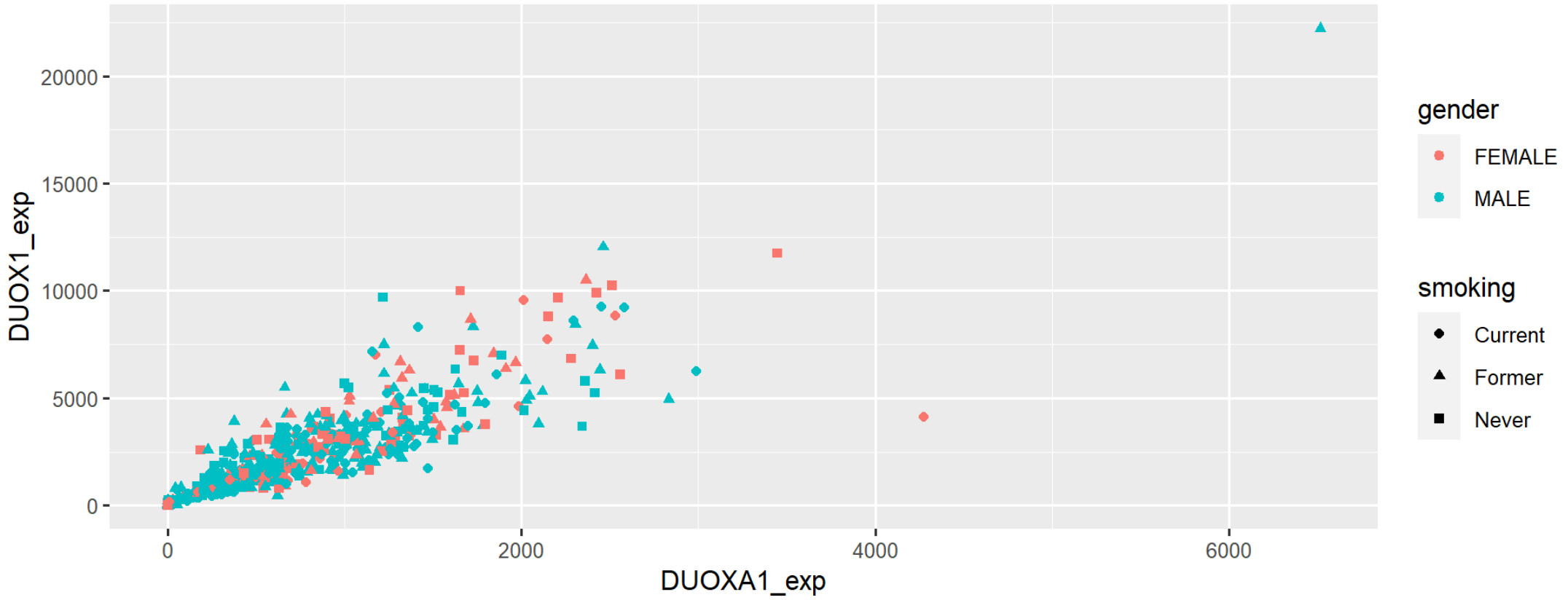
Scatter plots

R Code | Plot

```
ggplot(tcga %>% filter(!is.na(smoking))) +
  aes(DUOXA1_exp, DUOX1_exp) +
  geom_point(aes(shape = smoking, colour = gender))
```

Scatter plots

R Code | Plot



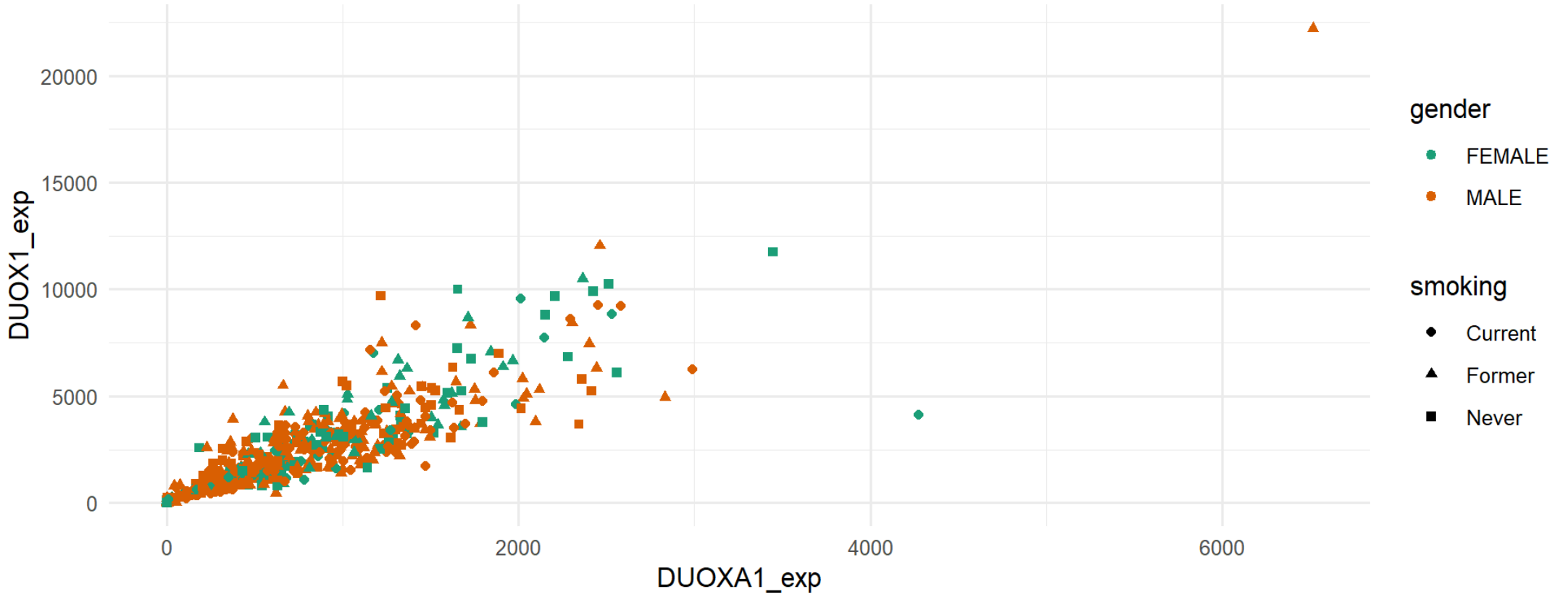
Scatter plots

R Code | Plot

```
ggplot(tcga %>% filter(!is.na(smoking))) +
  aes(DUOXA1_exp, DUOX1_exp) +
  geom_point(aes(shape = smoking, colour = gender))  +
  scale_color_brewer(palette="Dark2") +
  theme_minimal()
```

Scatter plots

R Code | Plot



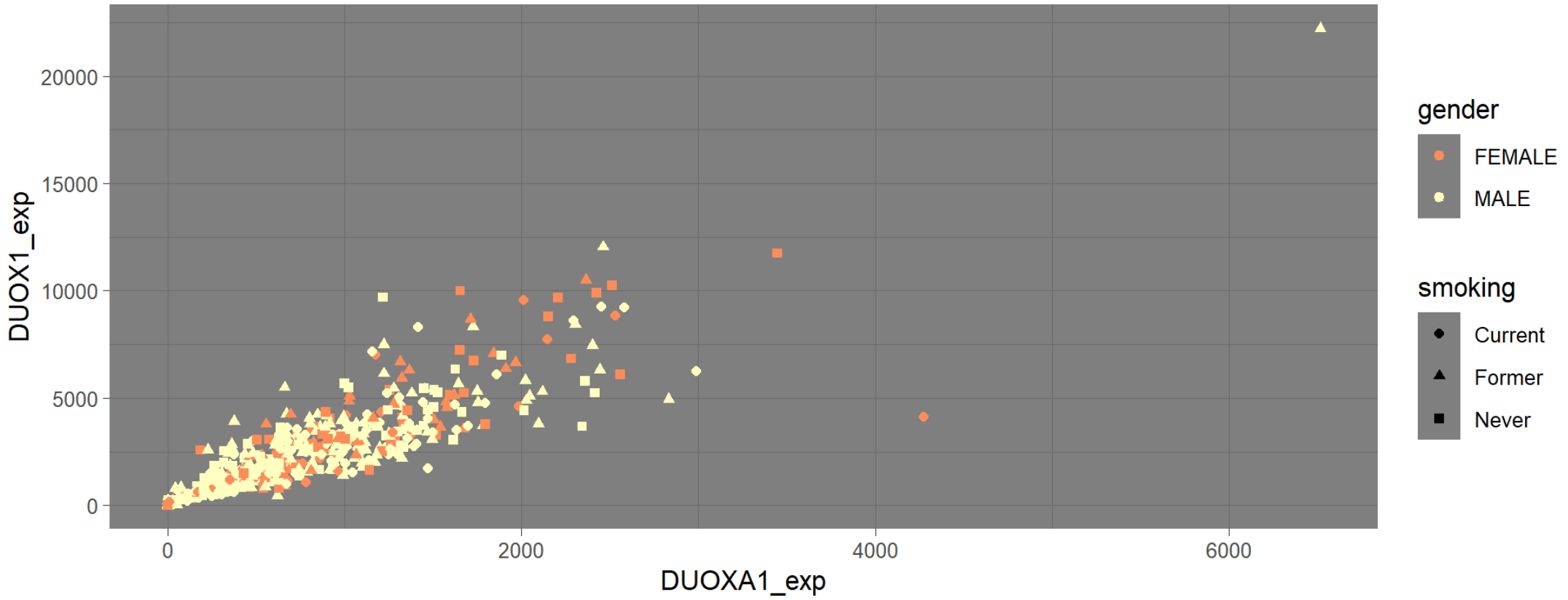
Scatter plots

R Code | Plot

```
ggplot(tcga %>% filter(!is.na(smoking))) +
  aes(DUOXA1_exp, DUOX1_exp) +
  geom_point(aes(shape = smoking, colour = gender))  +
  scale_color_brewer(palette="Spectral") +
  theme_dark()
```

Scatter plots

R Code | Plot



Getting help

- In R console

```
library(ggplot2)

?scale_color_brewer

?theme
```

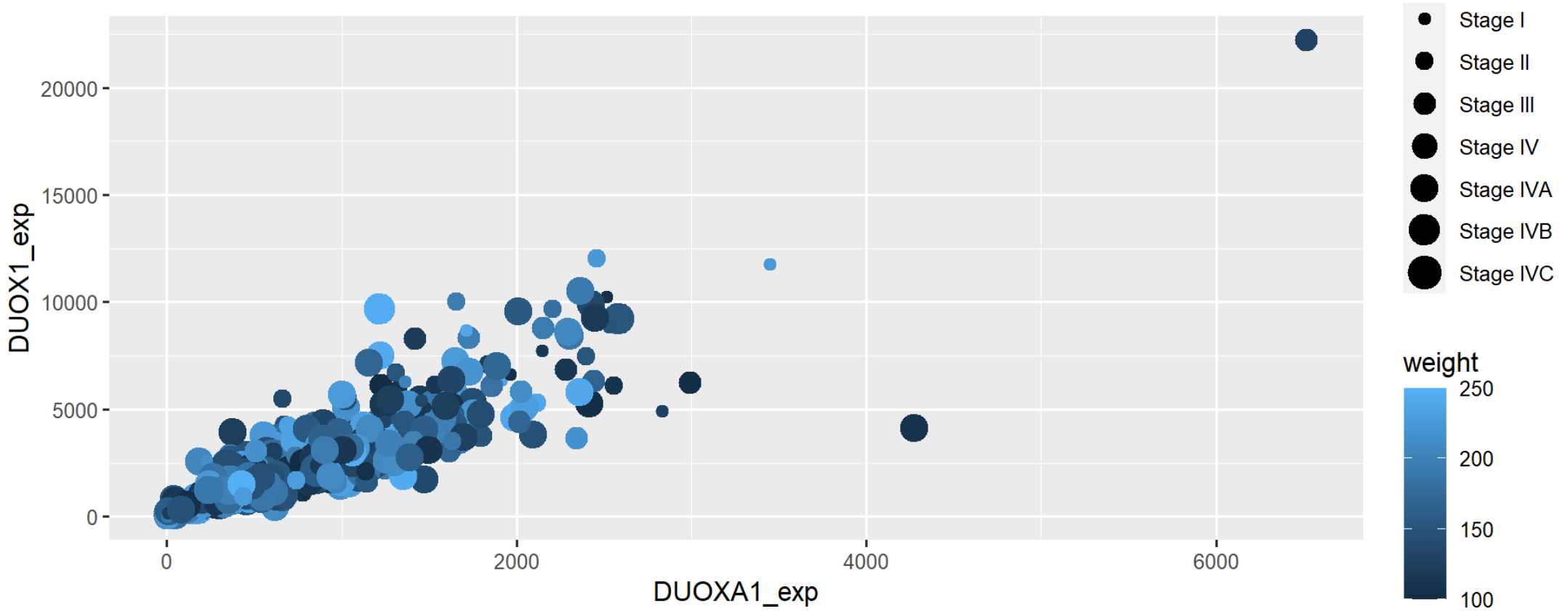
Scatter plots

R Code | Plot

```
ggplot(tcga) +
  aes(DUOXA1_exp, DUOX1_exp) +
  geom_point(aes(size = stage,colour = weight))
```

Scatter plots

R Code | Plot



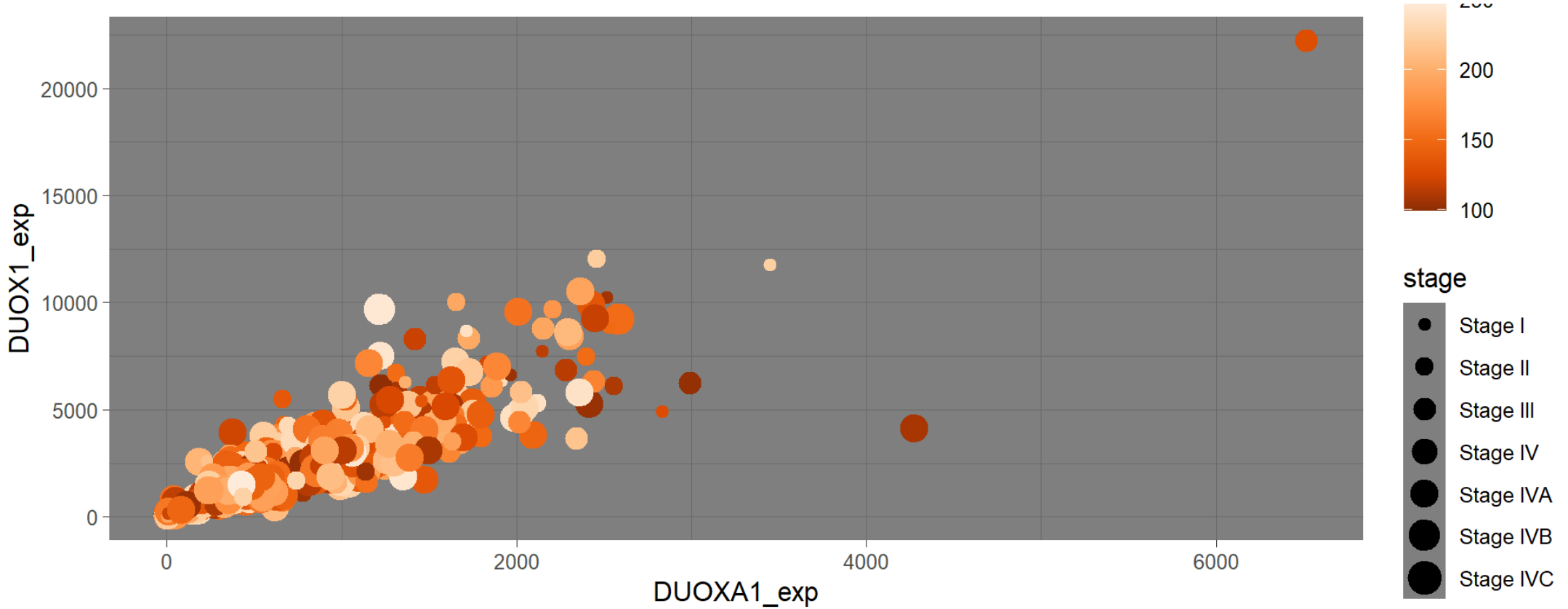
Scatter plots

R Code | Plot

```
ggplot(tcga) +
  aes(DUOXA1_exp, DUOX1_exp) +
  geom_point(aes(size = stage,colour = weight))   +
  scale_color_distiller(palette="Oranges") +
  theme_dark()
```

Scatter plots

R Code | Plot



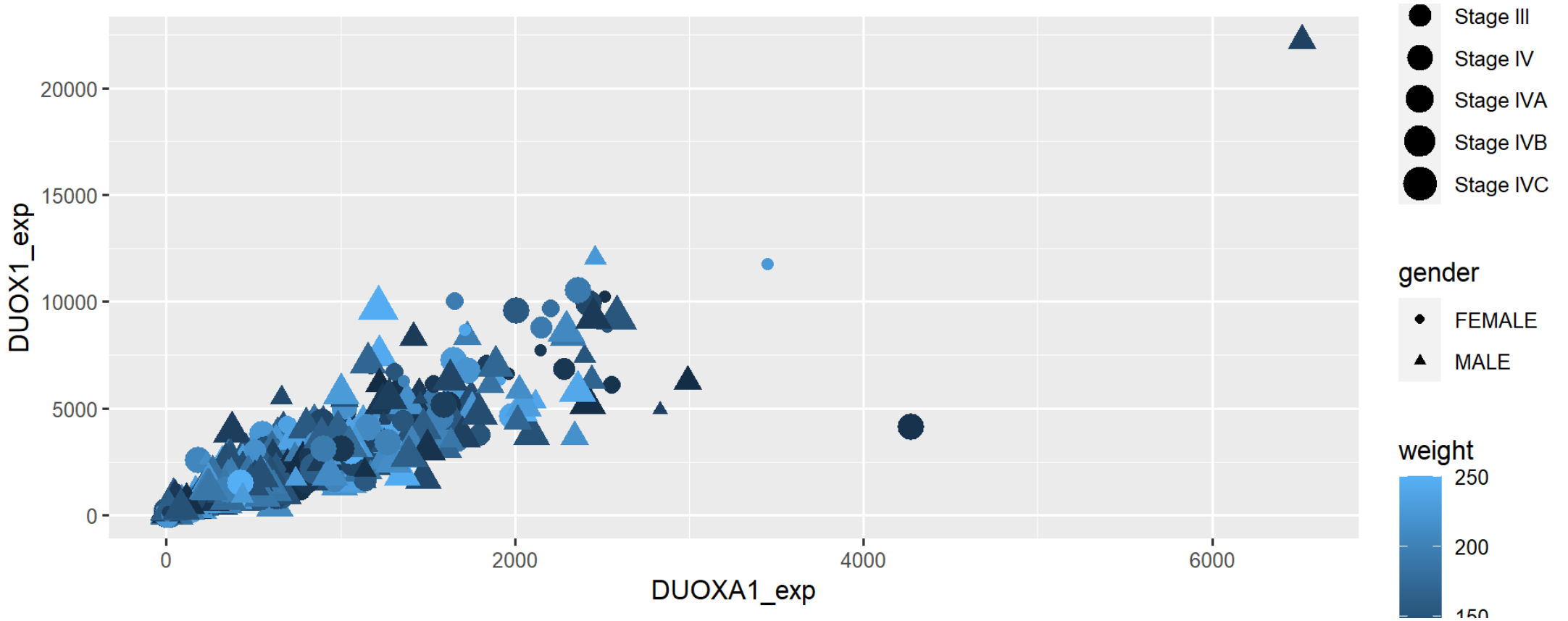
Scatter plots

R Code | Plot

```
ggplot(tcga) +
  aes(DUOXA1_exp, DUOX1_exp) +
  geom_point(aes(size = stage,colour = weight, shape = gender))
```

Scatter plots

R Code | Plot



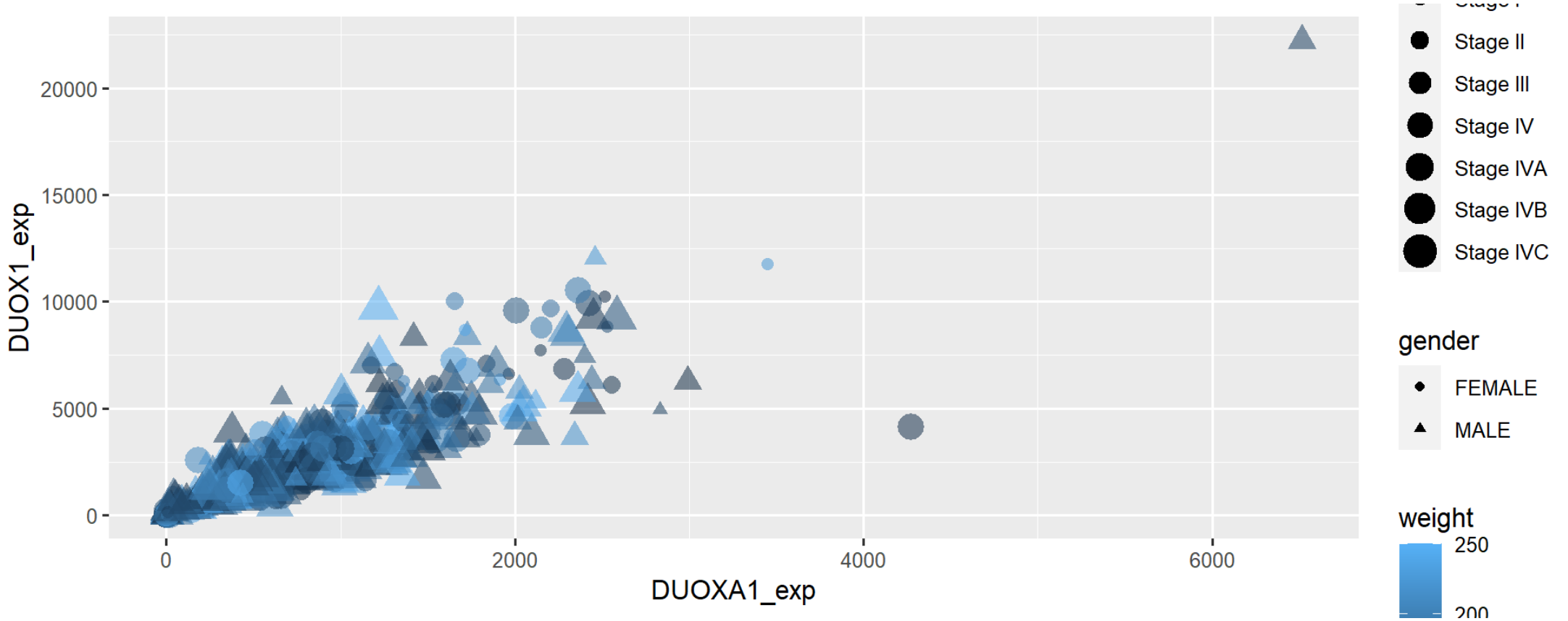
Scatter plots

R Code | Plot

```
ggplot(tcga) +
  aes(DUOXA1_exp, DUOX1_exp) +
  geom_point(aes(size = stage,colour = weight,
                 shape = gender, alpha = .5))
```

Scatter plots

R Code | Plot



Correlation

A correlation coefficient is a numerical measure of some type of correlation, meaning a statistical relationship between two variables.

Several types of correlation coefficient exist, each with their own definition and own range of usability and characteristics. They all assume values in the range from −1 to +1, where ±1 indicates the strongest possible agreement and 0 no agreement.

Very common :

Pearson &
Spearman

Pearson Correlation

The Pearson product-moment correlation coefficient, also known as r or Pearson's r, is a measure of the strength and direction of the linear relationship between two variables that is defined as the covariance of the variables divided by the product of their standard deviations. Pearson's r is the best-known and most commonly used type of correlation coefficient. A relationship is linear when a change in one variable is associated with a proportional change in the other variable.

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i-\overline{x})(y_i-\overline{y})}{\sqrt{(\sum_{i=1}^{n}(x_i-\overline{x})^2)}\sqrt{(\sum_{i=1}^{n}(y_i-\overline{y})^2)}}$$

- n is the sample size
- x_i and y_i are the individual sample points indexed with i
- $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the smaple mean of x (similarly y)

Pearson Correlation

```
r1 <- tcga %>% correlation::cor_test("DUOXA1_exp",
                                     "DUOX1_exp",
                                     method = c("pearson") )

r2 <- tcga %>% correlation::cor_test("BUB1_exp",
                                     "C10orf32_exp",
                                     method = c("pearson") )

r3 <- tcga %>% correlation::cor_test("BRAF_exp",
                                     "DTL_exp",
                                     method = c("pearson") )
```

Pearson Correlation

```
bind_rows(r1,r2,r3 )
```

Parameter1	Parameter2	r	95% CI	t(1041)	p
DUOXA1_exp	DUOX1_exp	0.92	[0.91, 0.93]	74.79	< .001***
BUB1_exp	C10orf32_exp	-0.60	[-0.64, -0.56]	-24.23	< .001***
BRAF_exp	DTL_exp	0.02	[-0.04, 0.08]	0.64	0.521

Observations: 1043

Pearson Correlation

```
knitr::kable(bind_rows(r1,r2,r3 ), format = 'html', digits = 3) %>%
 kable_styling(font_size = 12)
```

Parameter1	Parameter2	r	CI	CI_low	CI_high	t	df_error	p	Method	n_Obs
DUOXA1_exp	DUOX1_exp	0.918	0.95	0.908	0.927	74.789	1041	0.000	Pearson	1043
BUB1_exp	C10orf32_exp	-0.600	0.95	-0.638	-0.560	-24.227	1041	0.000	Pearson	1043
BRAF_exp	DTL_exp	0.020	0.95	-0.041	0.080	0.642	1041	0.521	Pearson	1043

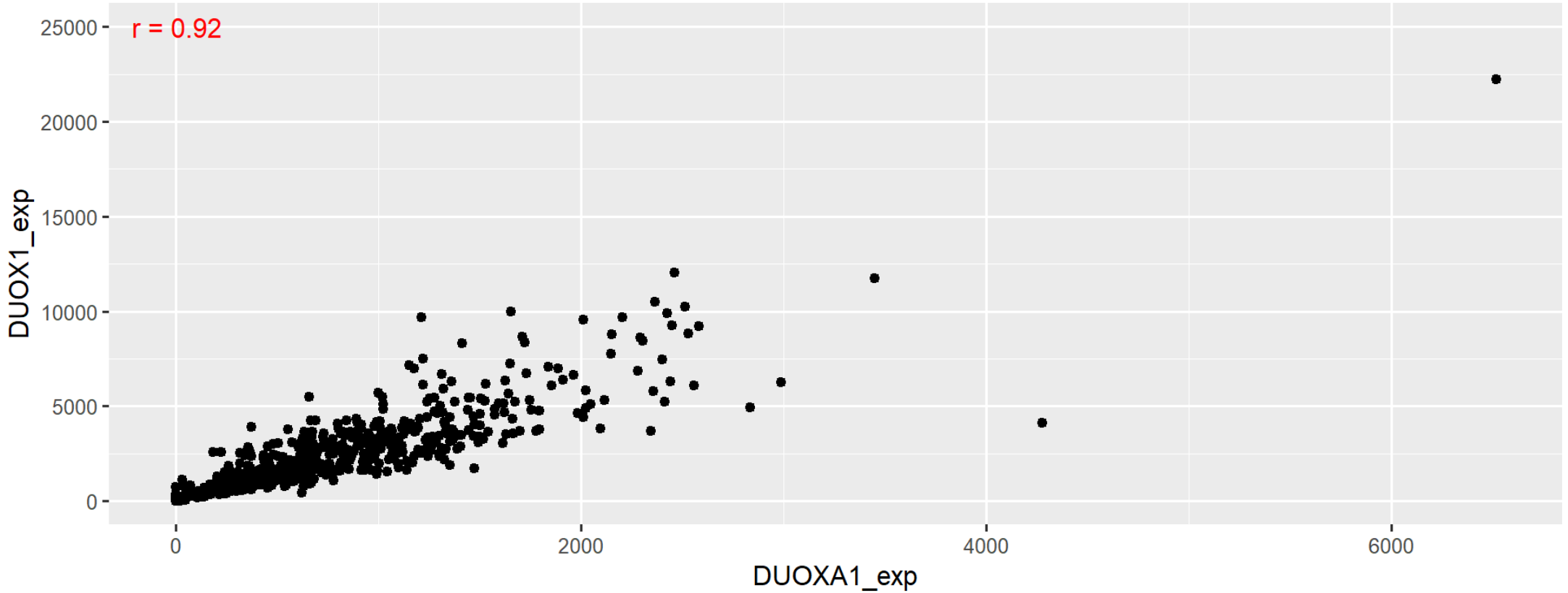
Pearson Correlation: Scatter Plot

R Code | Plot

```
ggplot(tcga) +
  aes(DUOXA1_exp, DUOX1_exp) +
  geom_point() +
  annotate(geom = "text", x = 10, y = 25000,
           label = paste("r = ", round(r1$r, 2), sep = ""),
           color = "red")
```

Pearson Correlation: Scatter Plot

R Code | Plot



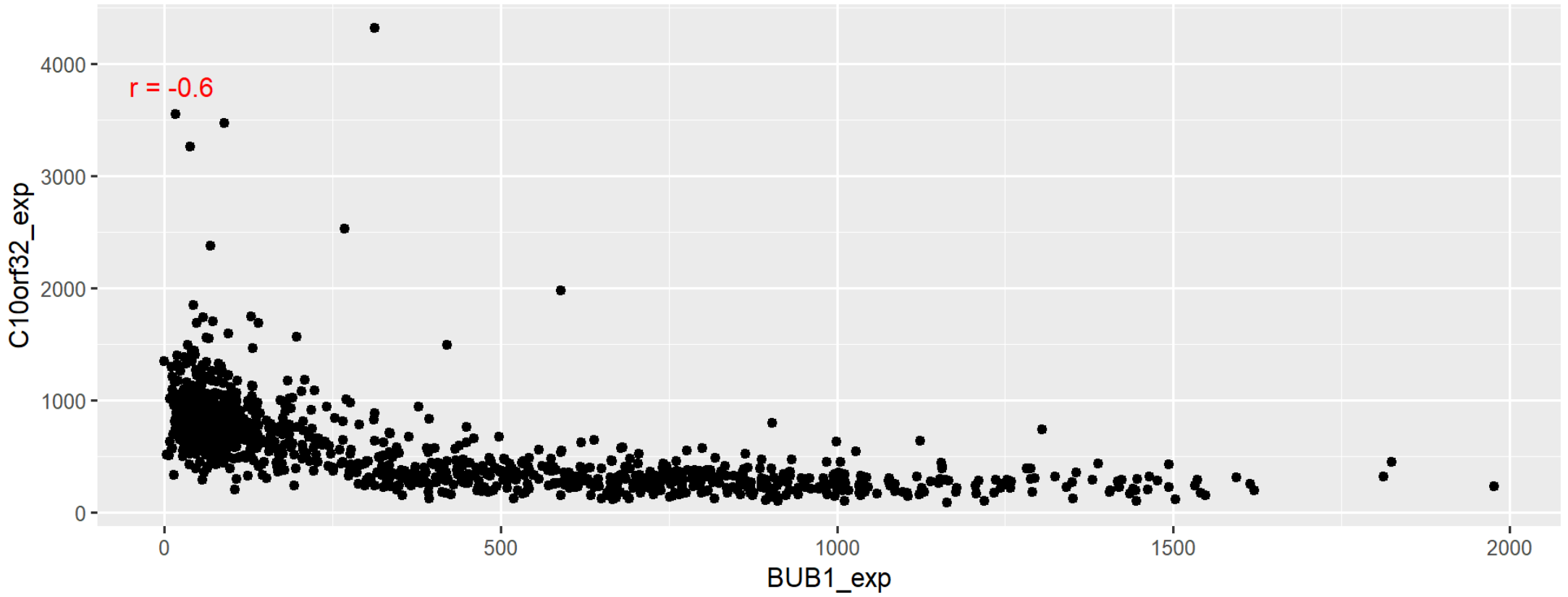
Pearson Correlation: Scatter Plot

R Code | Plot

```
ggplot(tcga) +
  aes(BUB1_exp, C10orf32_exp) +
  geom_point() +
  annotate(geom = "text", x = 10, y = 3800,
           label = paste("r = ", round(r2$r, 2), sep = ""),
           color = "red")
```

Pearson Correlation: Scatter Plot

R Code | Plot



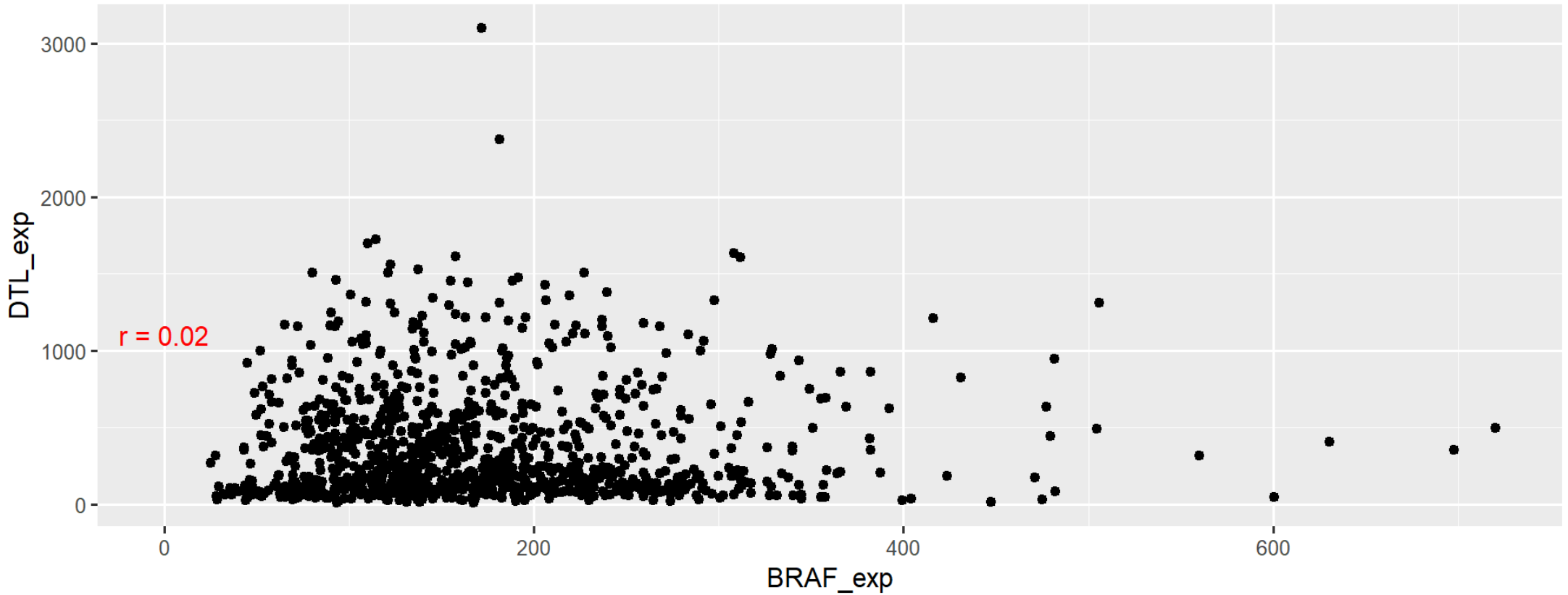
Pearson Correlation: Scatter Plot

R Code | Plot

```
ggplot(tcga) +
  aes(BRAF_exp, DTL_exp) +
  geom_point() +
  annotate(geom = "text", x = 0, y = 1100,
           label = paste("r = ", round(r3$r, 2), sep = ""),
           color = "red")
```

Pearson Correlation: Scatter Plot

R Code | Plot



Correlation Matrix

```
tcga %>%
  select(DUOXA1_exp, DUOX1_exp, BUB1_exp, BRAF_exp, DTL_exp ) %>%
  cor() %>% round(.,2)
```

	DUOXA1_exp	DUOX1_exp	BUB1_exp	BRAF_exp	DTL_exp
DUOXA1_exp	1.00	0.92	0.40	-0.07	0.30
DUOX1_exp	0.92	1.00	0.38	-0.05	0.30
BUB1_exp	0.40	0.38	1.00	0.01	0.78
BRAF_exp	-0.07	-0.05	0.01	1.00	0.02
DTL_exp	0.30	0.30	0.78	0.02	1.00

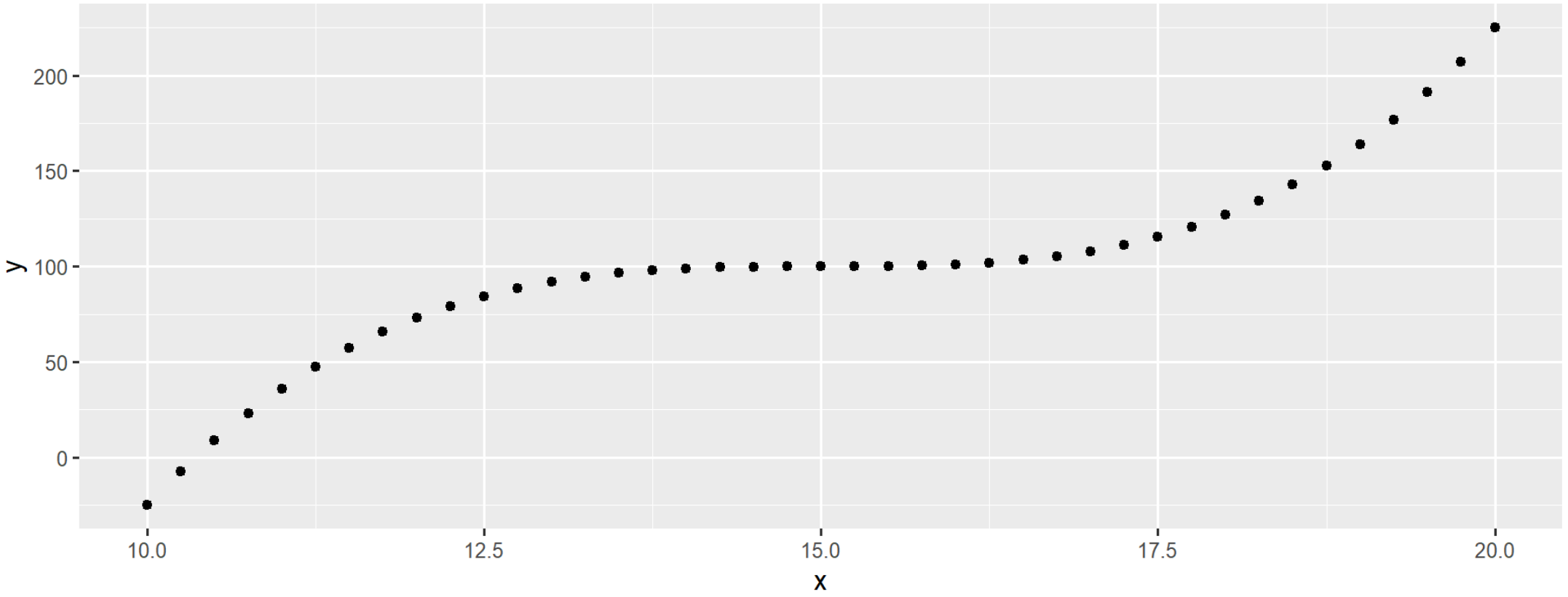
Spearman Correlation

The Spearman correlation evaluates the monotonic relationship between two continuous or ordinal variables. It is a nonparametric measure of rank correlation (statistical dependence between the rankings of two variables). In a monotonic relationship, the variables tend to change together, but not necessarily at a constant rate. The Spearman correlation coefficient is based on the ranked values for each variable rather than the raw data.

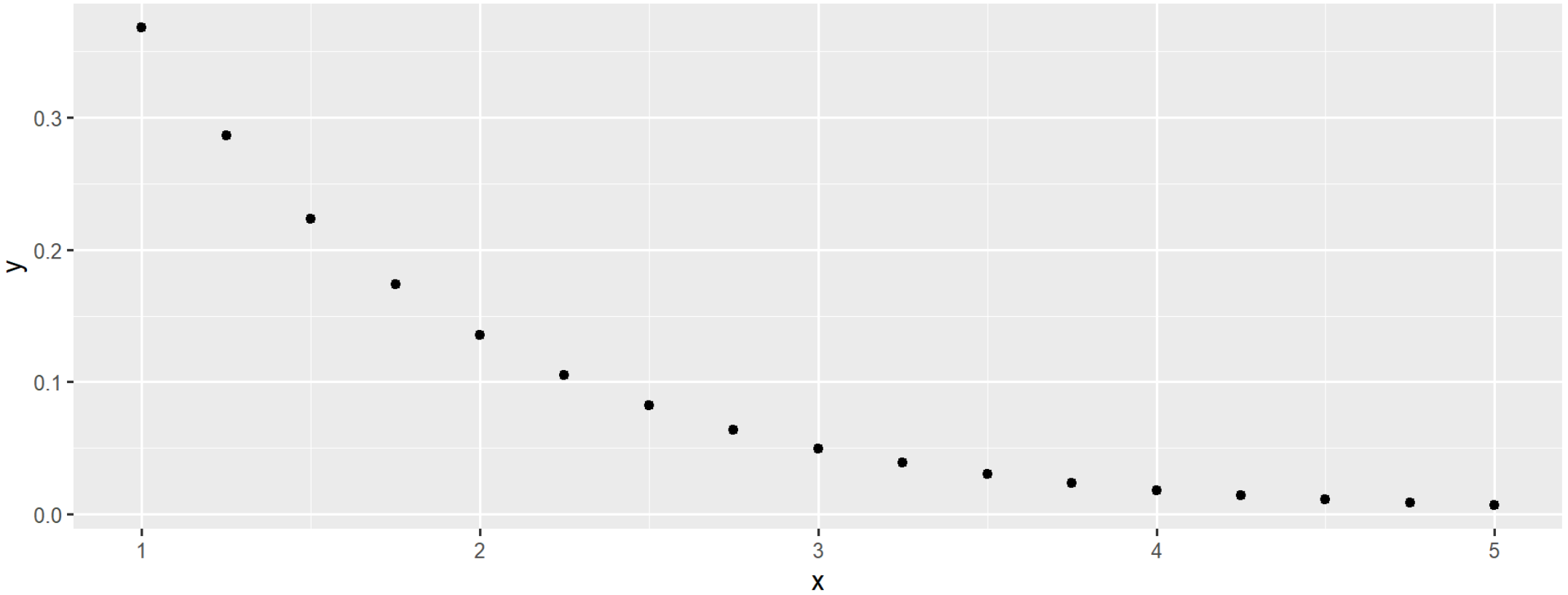
$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2-1)}$$

- d is the difference in ranks

Example of monotonic relationship



Example of monotonic relationship 2



Spearman Correlation

```
r1 <- tcga %>% correlation::cor_test("DUOXA1_exp",
                                     "DUOX1_exp",
                                     method = c("spearman") )

r2 <- tcga %>% correlation::cor_test("BUB1_exp",
                                     "C10orf32_exp",
                                     method = c("spearman") )

r3 <- tcga %>% correlation::cor_test("BRAF_exp",
                                     "DTL_exp",
                                     method = c("spearman") )
```

Spearman Correlation

```
knitr::kable(bind_rows(r1, r2, r3), format = 'html', digits = 3) %>%
 kable_styling(font_size = 12)
```

Parameter1	Parameter2	rho	CI	CI_low	CI_high	S	p	Method	n_Obs
DUOXA1_exp	DUOX1_exp	0.905	0.95	0.892	0.915	18053045	0.00	Spearman	1043
BUB1_exp	C10orf32_exp	-0.796	0.95	-0.818	-0.772	339618298	0.00	Spearman	1043
BRAF_exp	DTL_exp	0.015	0.95	-0.048	0.077	186283211	0.63	Spearman	1043

Pearson Correlation

```
bind_rows(r1,r2,r3 )
```

Parameter1	Parameter2	rho	95% CI	S	p
DUOXA1_exp	DUOX1_exp	0.90	[0.89, 0.92]	1.81e+07	< .001***
BUB1_exp	C10orf32_exp	-0.80	[-0.82, -0.77]	3.40e+08	< .001***
BRAF_exp	DTL_exp	0.01	[-0.05, 0.08]	1.86e+08	0.630

Observations: 1043

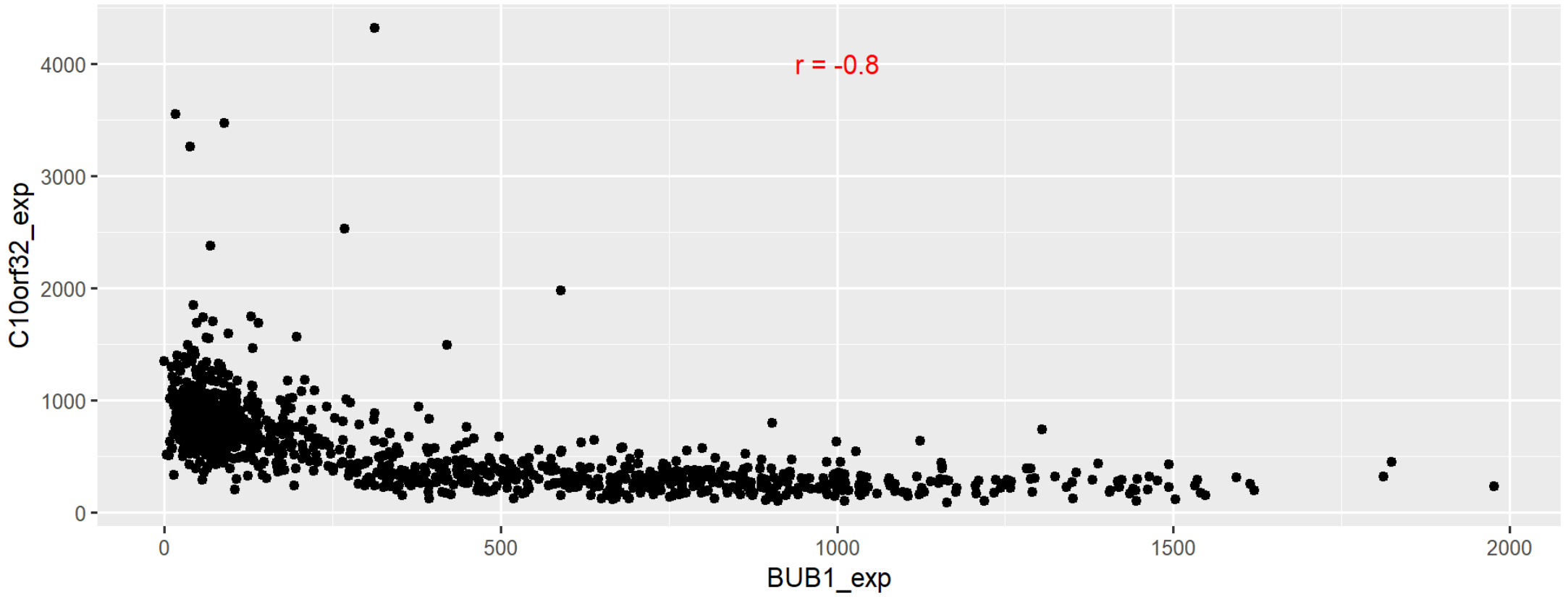
Spearman Correlation: Scatter Plot

R Code | Plot

```
ggplot(tcga) +
  aes(BUB1_exp, C10orf32_exp) +
  geom_point() +
  annotate(geom = "text", x = 1000, y = 4000,
           label = paste("r = ", round(r2$r, 2), sep = ""),
           color = "red")
```

Spearman Correlation: Scatter Plot

R Code | Plot



Correlation Matrix

```
tcga %>%
  select(DUOXA1_exp, DUOX1_exp, BUB1_exp, BRAF_exp, DTL_exp ) %>%
  cor( method = c("spearman")) %>% round(.,2)
```

	DUOXA1_exp	DUOX1_exp	BUB1_exp	BRAF_exp	DTL_exp
DUOXA1_exp	1.00	0.90	0.67	-0.11	0.62
DUOX1_exp	0.90	1.00	0.66	-0.10	0.62
BUB1_exp	0.67	0.66	1.00	-0.01	0.88
BRAF_exp	-0.11	-0.10	-0.01	1.00	0.01
DTL_exp	0.62	0.62	0.88	0.01	1.00

Regression

- Univariate regresssion
- MultivarIABLE reression

Univariate Regression

Simple linear regression: 1 predictor

$Y_i = \beta_0 + \beta_1 X_1 + \epsilon_i$ where $\epsilon_i \sim N(0, \sigma^2)$

- X_1 is the predictor or independent variable.
- β_0 is the intercept and β_1 is the slope
- ϵ is the error vector (residuals)
- Y_i is the response vector or dependent variable

Regression assumptions

There are four assumptions associated with a linear regression model:

- Linearity: The relationship between X and Y is linear.
- Independence: Observations are independent of each other.
- Homoscedasticity: The variance of residuals is the same for any value of X.
- Normality: For any fixed value of X, Y is normally distributed.

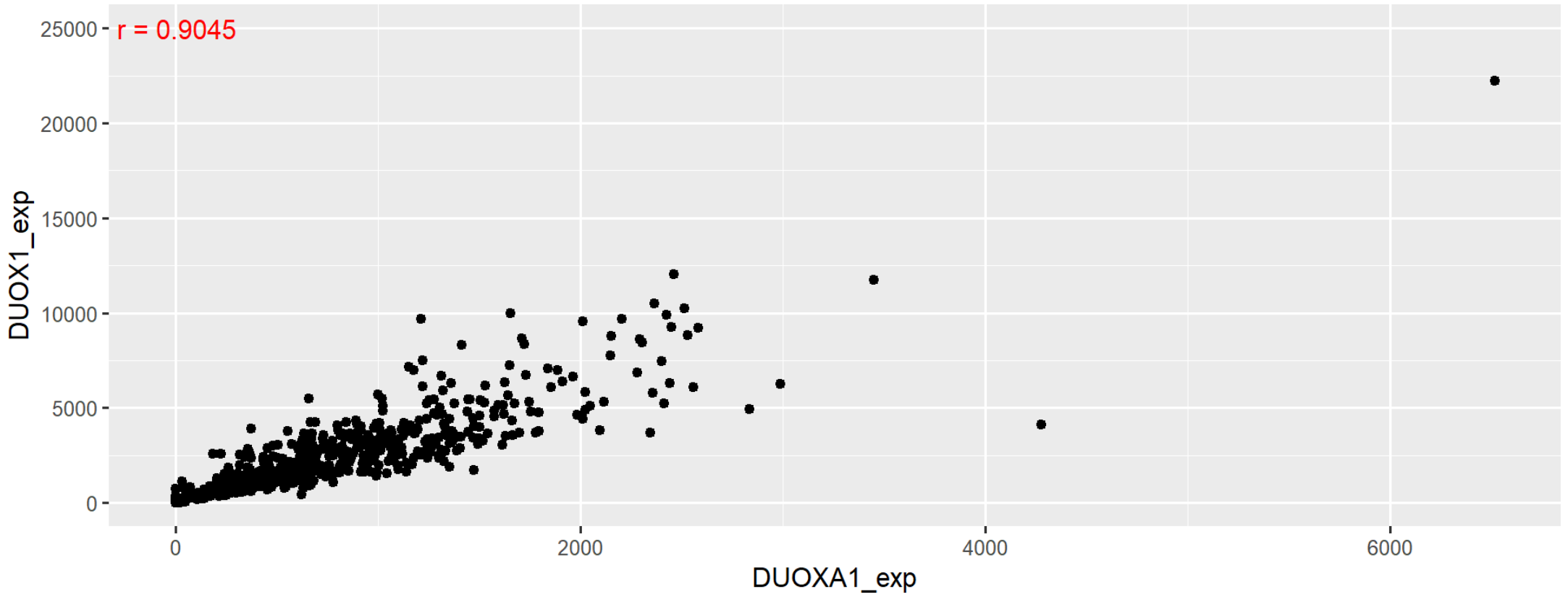
Scatter Plot

R Code | Plot

```
ggplot(tcga) +
  aes(DUOXA1_exp, DUOX1_exp) +
  geom_point() +
  annotate(geom = "text", x = 10, y = 25000,
           label = paste("r = ", round(r1$r, 4), sep = ""),
           color = "red")
```

Scatter Plot

R Code | Plot



Simple linear regression

```
lm(DUOX1_exp ~ DUOXA1_exp, data = tcga)
```

```
Call:
lm(formula = DUOX1_exp ~ DUOXA1_exp, data = tcga)

Coefficients:
(Intercept)   DUOXA1_exp
     84.712        3.034
```

Simple linear regression

```
summary( lmfit <-  lm(DUOX1_exp ~ DUOXA1_exp, data = tcga) )
```

```
Call:
lm(formula = DUOX1_exp ~ DUOXA1_exp, data = tcga)

Residuals:
    Min      1Q  Median      3Q     Max 
-8924.8   -85.3   -70.7    41.8  5917.1 

Coefficients:
             Estimate Std. Error t value             Pr(>|t|)    
(Intercept) 84.71188   31.04085   2.729              0.00646 ** 
DUOXA1_exp   3.03400    0.04057  74.789 < 0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 824.9 on 1041 degrees of freedom
Multiple R-squared:  0.8431,	Adjusted R-squared:  0.8429 
```

Simple linear regression

Residual standard error: 824.9 on 1041 degrees of freedom

Multiple R-squared: 0.8431, Adjusted R-squared: 0.8429

F-statistic: 5593 on 1 and 1041 DF, p-value: < 0.00000000000000022

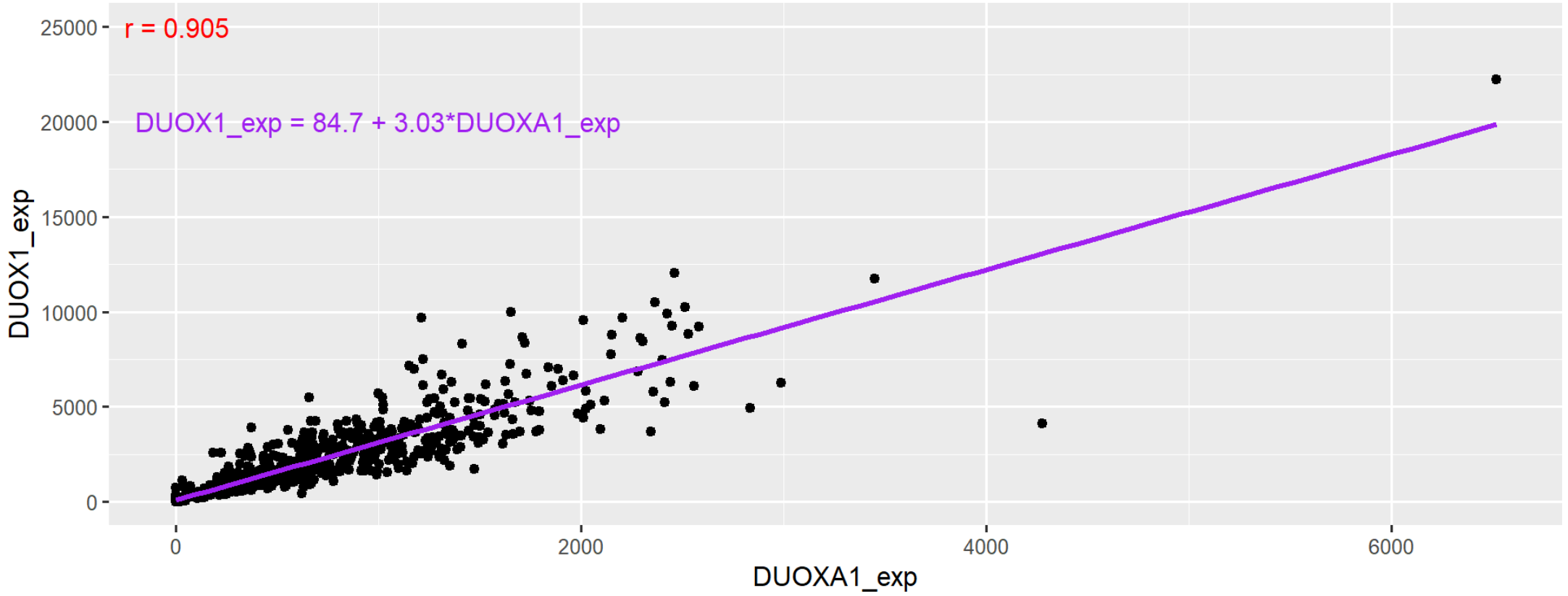
Scatter Plot with linear regression line

R Code | Plot

```
ggplot(tcga, aes(DUOXA1_exp, DUOX1_exp )) +
  geom_point() +
  geom_smooth(method = "lm", se=FALSE, color="purple", formula = y ~ x,
  annotate(geom = "text", x = 10, y = 25000,
           label = paste("r = ", round(r1$r, 3), sep = ""),
           color = "red") +
  annotate(geom = "text", x = 1000, y = 20000,
           label = paste("DUOX1_exp = 84.7 + 3.03*DUOXA1_exp", sep = ""
           color = "purple")
```

Scatter Plot with linear regression line

R Code | Plot



Zoom in on the Scatter Plot with linear regression line

R Code | Plot

```
ggplot(tcga, aes(DUOXA1_exp, DUOX1_exp)) +
  geom_point() +
  geom_smooth(method = "lm", se=FALSE, color="purple", formula = y ~ x,
  annotate(geom = "text", x = 10, y = 25000,
           label = paste("r = ", round(r1$r, 2), sep = ""), color = "re
  annotate(geom = "text", x = 1000, y = 20000,
           label = paste("DUOX1_exp = 84.7 + 3.03*DUOXA1_exp", sep = ""
  coord_cartesian(ylim=c(0, 500), xlim=c(0, 250))
```

Zoom in on the Scatter Plot with linear regression line

R Code | Plot

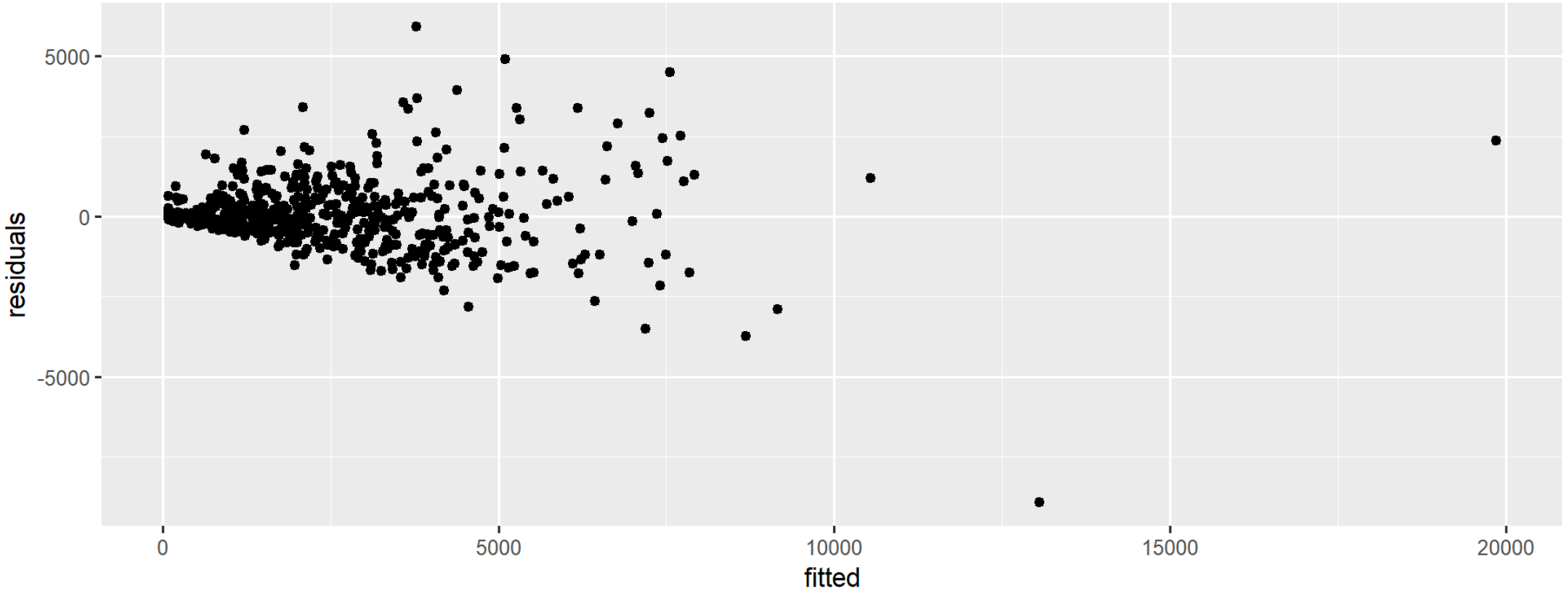
Residual Scatter Plot

R Code | Plot

```
resid <- data.frame(fitted = lmfit$fitted.values , residuals = lmfit$res
ggplot( resid, aes(fitted, residuals) ) +
    geom_point()
```

Residual Scatter Plot

R Code | Plot



Simple linear regression - sqrt transformation

```
summary(lmfit <- lm(sqrt(DUOX1_exp) ~ sqrt(DUOXA1_exp), data = tcga))
```

```
Call:
lm(formula = sqrt(DUOX1_exp) ~ sqrt(DUOXA1_exp), data = tcga)

Residuals:
    Min      1Q  Median      3Q     Max
-46.781  -2.707  -0.760   2.286  37.547

Coefficients:
                 Estimate Std. Error t value             Pr(>|t|)
(Intercept)       3.57802    0.29514   12.12 <0.0000000000000002 ***
sqrt(DUOXA1_exp)  1.64382    0.01415  116.14 <0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Simple linear regression - sqrt transformation

Residual standard error: 6.94 on 1041 degrees of freedom

Multiple R-squared: 0.9284, Adjusted R-squared: 0.9283

F-statistic: 1.349e+04 on 1 and 1041 DF, p-value: < 0.00000000000000022

Simple linear regression - sqrt transformation

```
names( lmfit )
```

```
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"
```

```
names( summary(lmfit) )
```

```
 [1] "call"          "terms"         "residuals"     "coefficients"
 [5] "aliased"       "sigma"         "df"            "r.squared"
 [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

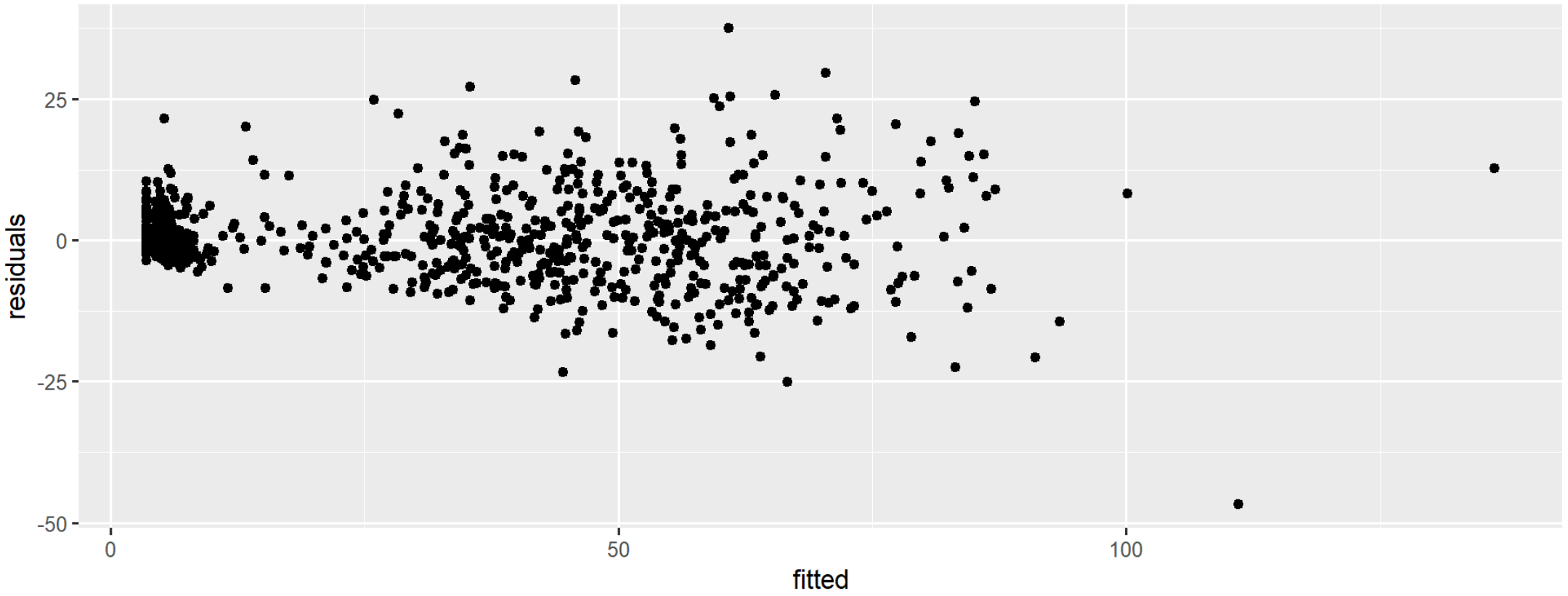
Residual Scatter Plot - sqrt transformation

R Code | Plot

```
resid <- data.frame(fitted = lmfit$fitted.values , residuals = lmfit$res
ggplot( resid, aes(fitted, residuals) ) +
    geom_point()
```

Residual Scatter Plot - sqrt transformation

R Code | Plot



Scatter Plot with linear regression line - sqrt transformation

R Code | Plot

```
ggplot(tcga, aes(sqrt(DUOXA1_exp), sqrt(DUOX1_exp) )) +
  geom_point() +
  geom_smooth(method = "lm", se=FALSE, color="purple", formula = y ~ x,

  annotate(geom = "text", x = 50, y = 1.75,
           label = paste("sqrt(DUOX1_exp) = 3.58 + 1.64*srqt(DUOXA1_exp)
           color = "purple")
```

Scatter Plot with linear regression line - sqrt transformation

R Code | Plot

Multiple Regression

More than one predictor

$Y_i = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k + \epsilon_i$ for i = 1, 2, ..., n

$\epsilon_i \sim N(0, \sigma^2)$

- Y_i is the value of the response for the ith case
- $\epsilon_i \sim N(0, \sigma^2)$ (as before)
- $X_{i,k}$ is the value of the *k* th explanatory variable for the *i* th case.
- β_0 is the intercept
- $\beta_1, \beta_2, \ldots \beta_k$ are the regression coefficients for the explanatory variables
- ϵ is the error vector (residuals)
- Parameters as usual include all of the β's as well as σ^2. These need to be estimated from the data.

Multiple Regression

More than one predictor

$Y_i = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k + \epsilon_i$ for i = 1, 2, ..., n

- Here we're using *k* for the number of predictor variables, which means we have *k*+1 regression parameters (the β coefficients).
- We assume that the ϵ_i have a normal distribution with mean 0 and constant variance σ^2. These are the same assumptions that we used in simple regression with one predictor.
- The subscript *i* refers to the ith individual or unit in the population. In the notation for the predictor.s, the subscript following *i* simply denotes which predictor it is.
- The word "linear" in "multiple linear regression" refers to the fact that the model is linear in the parameters, $\beta_0, \beta_1, \beta_2, \beta_k$. This means that each parameter multiplies a

Multiple regression

```
lm(DUOX1_exp ~ DUOXA1_exp + gender, data = tcga)
```

```
Call:
lm(formula = DUOX1_exp ~ DUOXA1_exp + gender, data = tcga)

Coefficients:
(Intercept)   DUOXA1_exp   genderMALE
    149.385        3.033      -93.292
```

Multiple regression

```
summary( lm(DUOX1_exp ~ DUOXA1_exp  + gender, data = tcga) )
```

```
Call:
lm(formula = DUOX1_exp ~ DUOXA1_exp + gender, data = tcga)

Residuals:
    Min      1Q  Median      3Q     Max
-8986.8  -144.2   -47.9    49.4  5946.5

Coefficients:
             Estimate Std. Error t value             Pr(>|t|)
(Intercept) 149.38510   49.25297   3.033              0.00248 **
DUOXA1_exp    3.03338    0.04053  74.837 < 0.0000000000000002 ***
genderMALE  -93.29234   55.19471  -1.690              0.09128 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 824.1 on 1040 degrees of freedom
```

Thank you!

- The end