project_phase1

```
library(data.table)
library(e1071)
library(ggplot2)
library(corrplot)
library(ggmosaic)
```

Q0-

```
# dota2_table = data.table(read.csv(file="C:\\Users\\Hamed\\Desktop\\DOTA2\\D
OTA2.csv", header=TRUE, sep=","))
dota2 = fread("C:\\Users\\Hamed\\Desktop\\DOTA2\\DOTA2.csv")
N = nrow(dota2)
apply(is.na(dota2),2,sum)/N
```

```
##                         V1                   match_id
##               0.0000000000               0.0000000000
##                 start_time                   duration
##               0.0000000000               0.0000000000
##      tower_status_radiant         tower_status_dire
##               0.0000000000               0.0000000000
##      barracks_status_dire   barracks_status_radiant
##               0.0000000000               0.0000000000
##          first_blood_time                  game_mode
##               0.0000000000               0.0000000000
##                radiant_win                       date
##               0.0000000000               0.0000000000
##                 account_id                    hero_id
##               0.0000000000               0.0000000000
##                player_slot                       gold
##               0.0000000000               0.0000000000
##                 gold_spent               gold_per_min
##               0.0000000000               0.0000000000
##                 xp_per_min                      kills
##               0.0000000000               0.0000000000
##                     deaths                    assists
##               0.0000000000               0.0000000000
##                     denies                  last_hits
##               0.0000000000               0.0000000000
##                hero_damage               hero_healing
##               0.0000000000               0.0000000000
##               tower_damage                      level
##               0.0000000000               0.0000000000
##                    xp_hero                   xp_creep
##               0.0018294132               0.0000000000
##                  xp_roshan                 gold_death
##               0.6330244714               0.0100736517
```

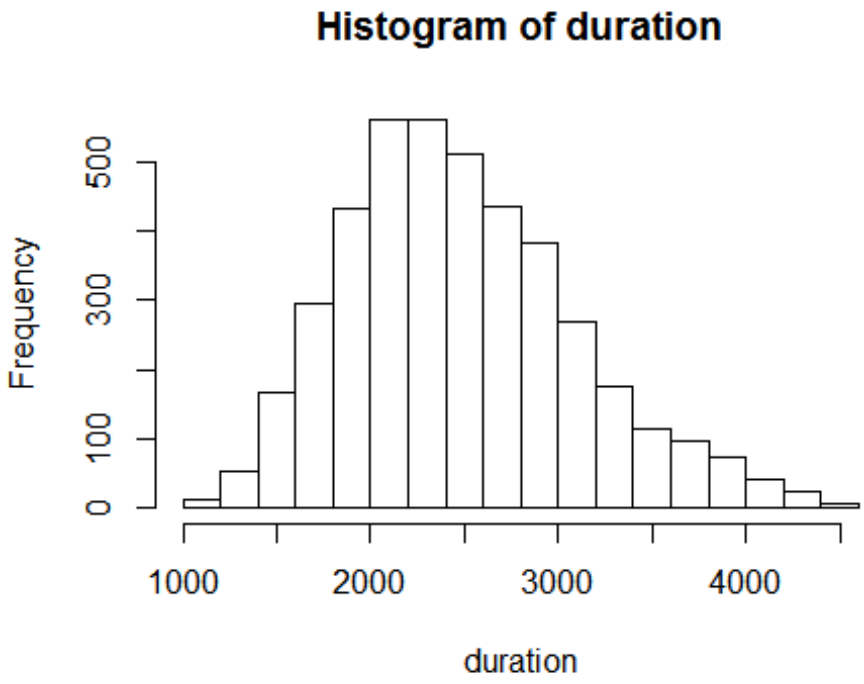
```
## gold_destroying_structure        gold_killing_heros
##               0.0348063673              0.0011879306
##       gold_killing_creeps       gold_killing_roshan
##               0.0001425517              0.4710144928
##          unit_order_total                      team
##               0.0000000000              0.0000000000
##                      Name                 All_Roles
##               0.0000000000              0.0000000000
##                  Carry_Car              Disabler_Dis
##               0.0000000000              0.0000000000
##              Initiator_Ini               Jungler_Jun
##               0.0000000000              0.0000000000
##               Support_Sup               Durable_Dur
##               0.0000000000              0.0000000000
##                  Nuker_Nuk                 Pusher_Pus
##               0.0000000000              0.0000000000
##                 Escape_Esc                 Role_Count
##               0.0000000000              0.0000000000
##                       Class                         WL
##               0.0000000000              0.0000000000
##                        Win
##               0.0000000000
```

Handling missing values depends on the statistics we want to measure. For example, if we want to compute median or mode we can simply ignore it but in order to compute mean it's better to predict its value using proper approaches.

Q1-

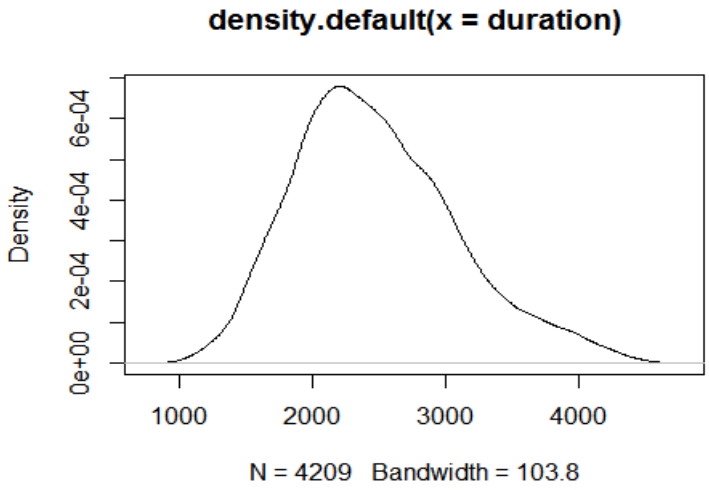
1-

```
duration = dota2[!duplicated(dota2[,'match_id']),'duration']$duration
K = 1 + 3.322*log10(length(duration))
hist(duration,breaks = K)
```




```
plot(density(duration))
```

2-



3- As mean is greater than median the distribution is right skewed, also we have one maximum so distribution is unimodal.

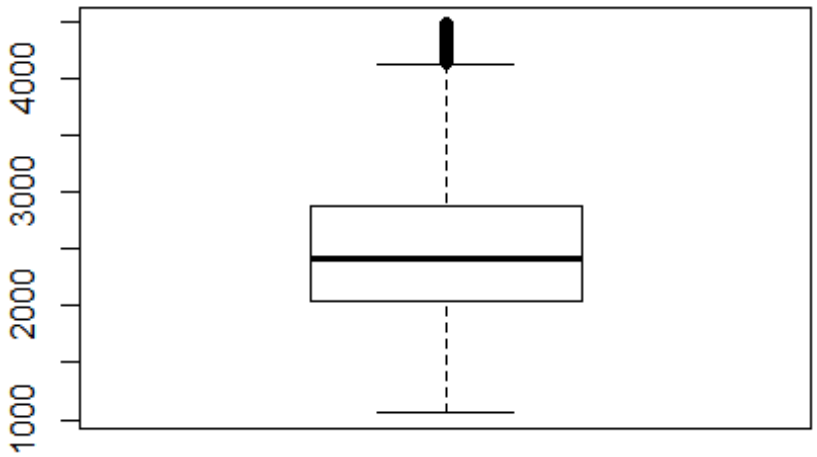
4-

```r
print(sprintf("mean:%f , median:%f , variance:%f , standard deviation:%f , skewness:%f",mean(duration),median(duration),sd(duration)^2,sd(duration),skewness(duration)))
```

```
## [1] "mean:2481.217154 , median:2406.000000 , variance:374608.309772 , standard deviation:612.052538 , skewness:0.531086"
```

5-

```r
boxplot(duration)
```




```r
print(sprintf("lower quartile:[%d,%#d]",quantile(duration)[1],quantile(duration)[2]))
```

```
## [1] "lower quartile:[1064,2036]"
```

```r
print(sprintf("upper quartile:[%d,%d]",quantile(duration)[4],quantile(duration)[5]))
```

```
## [1] "upper quartile:[2869,4477]"
```

```r
IQR = quantile(duration)[4] - quantile(duration)[2]
print(sprintf("IQR: %d",IQR))
```

```
## [1] "IQR: 833"
```

```r
print(sprintf("lower_inner_face: %#.2f",quantile(duration)[2]- 1.5*IQR))
```

```
## [1] "lower_inner_face: 786.50"
```

```
print(sprintf("upper_outer_face: %#.2f",quantile(duration)[4]+ 3*IQR))

## [1] "upper_outer_face: 5368.00"

 #Or use IQR(data)
```

6- As it's shown on the boxplot outliers are variables greater than lower inner face which is approximately 4000.

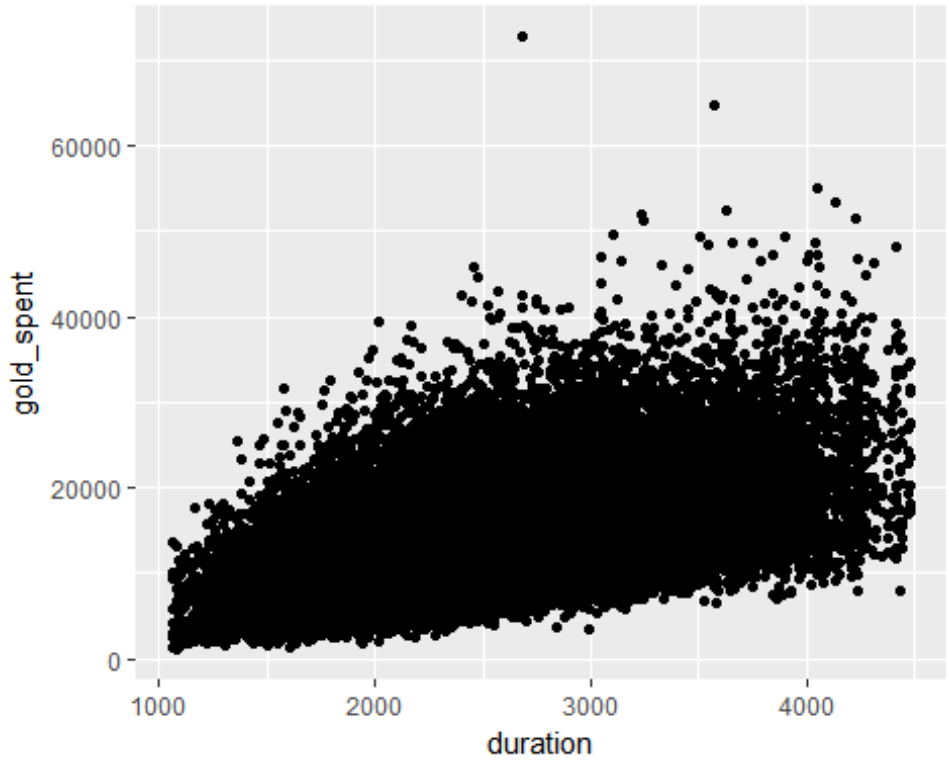
```
lower_inner_face = quantile(duration)[2]- 1.5*IQR
lower_outer_face = quantile(duration)[2]- 3*IQR
upper_inner_face = quantile(duration)[4]+ 1.5*IQR
upper_outer_face = quantile(duration)[4]+ 3*IQR
mid_outilers = duration[duration > upper_inner_face | duration < lower_inner_
face]
extreme_outilers = duration[duration > upper_outer_face | duration < lower_ou
ter_face]
```

As it's expected there are no extreme outliers and mild outliers are more than 4118.

Q2-

1-

```
ggplot(dota2, aes(x=duration, y=gold_spent)) + geom_point()
```




```
# Change the point size, and shape
```

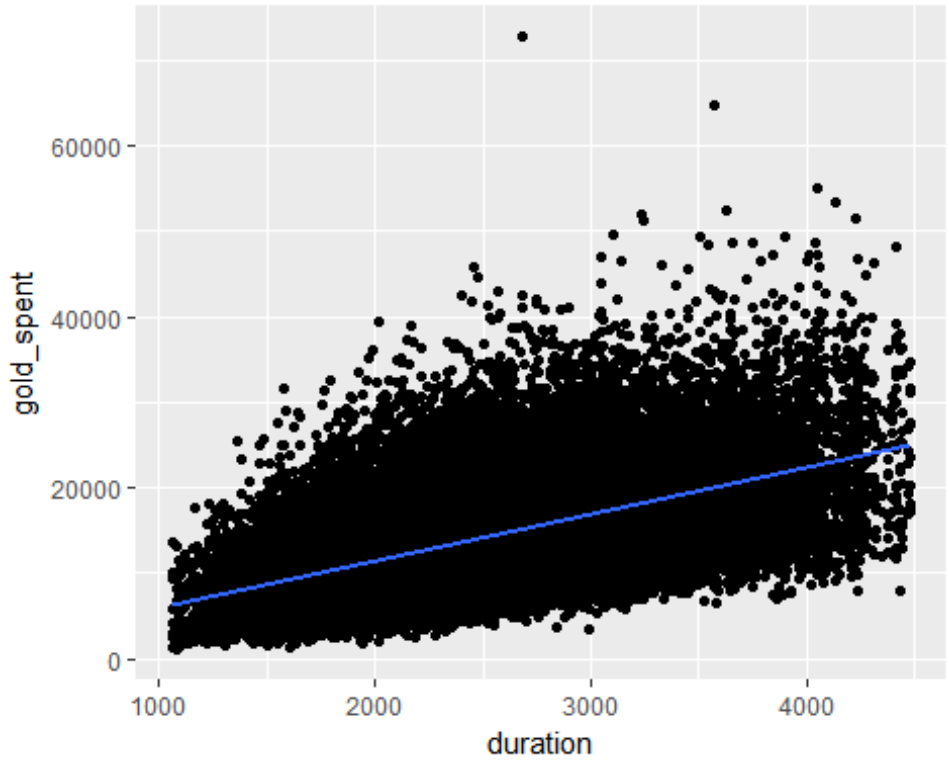
we plot amount of gold a player spent during match versus match duration. As it's expected the longer a match is ,higher amount of gold is spent commonly.
2-

```
cor(dota2$duration,dota2$gold_spent)

## [1] 0.5330669
```

3-

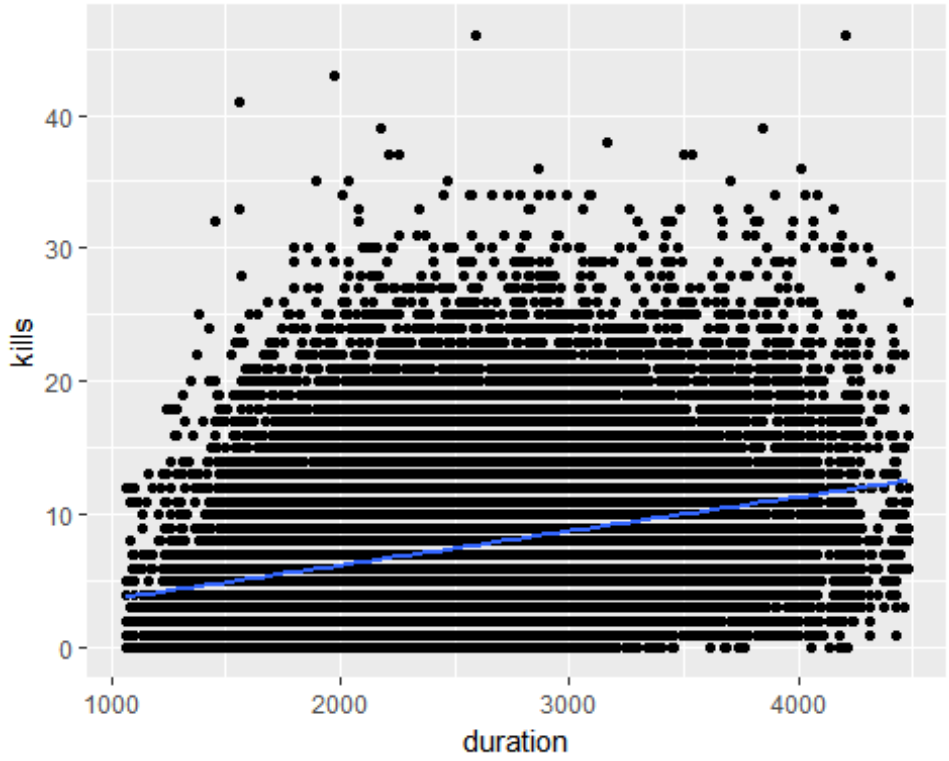
```
ggplot(dota2, aes(x=duration, y=gold_spent)) + geom_point() + geom_smooth(met
hod=lm)
```



As it's shown on the plot, line's slope is positive which agrees with positive correlation coefficient. But the value of slope doesn't tell us anything about value of correlation coefficient.

4- first pair:

```
ggplot(dota2, aes(x=duration, y=kills)) + geom_point() + geom_smooth(method=l
m)
```

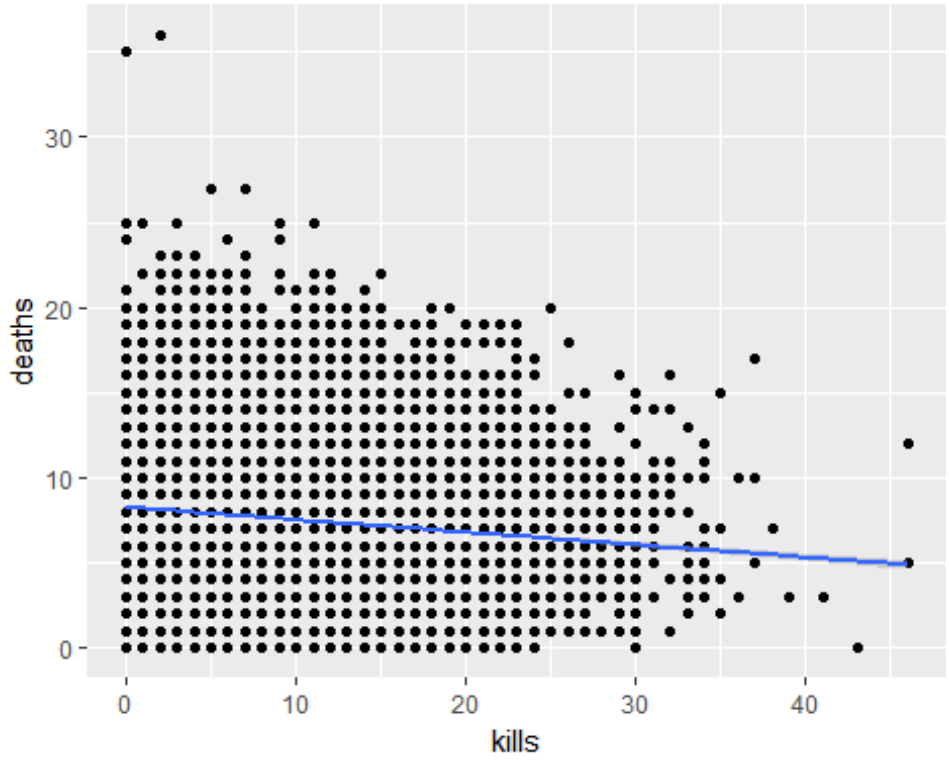



```
cor(dota2$duration,dota2$kills)
```

```
## [1] 0.288847
```

As it can be seen there's no strong correlation between a player's kills and match duration but it's somehow positively correlated. Second pair:

```
ggplot(dota2, aes(x=kills, y=deaths)) + geom_point() + geom_smooth(method=lm)
```



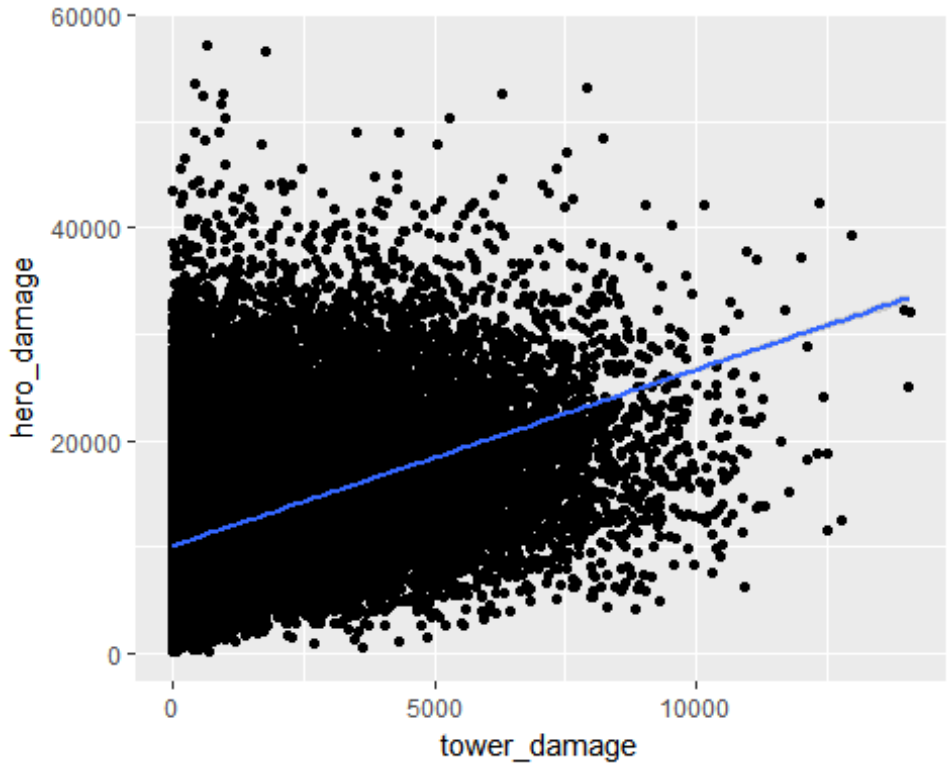

```
cor(dota2$deaths,dota2$kills)

## [1] -0.105592
```

Again there's no strong correlation between a player's kills and his deaths in a match but it's somehow negatively correlated. which means that players who kill might be killed less.

Third pair:

```
ggplot(dota2, aes(x=tower_damage, y=hero_damage)) + geom_point() + geom_smoot
h(method=lm)
```




```
cor(dota2$tower_damage,dota2$hero_damage)
```

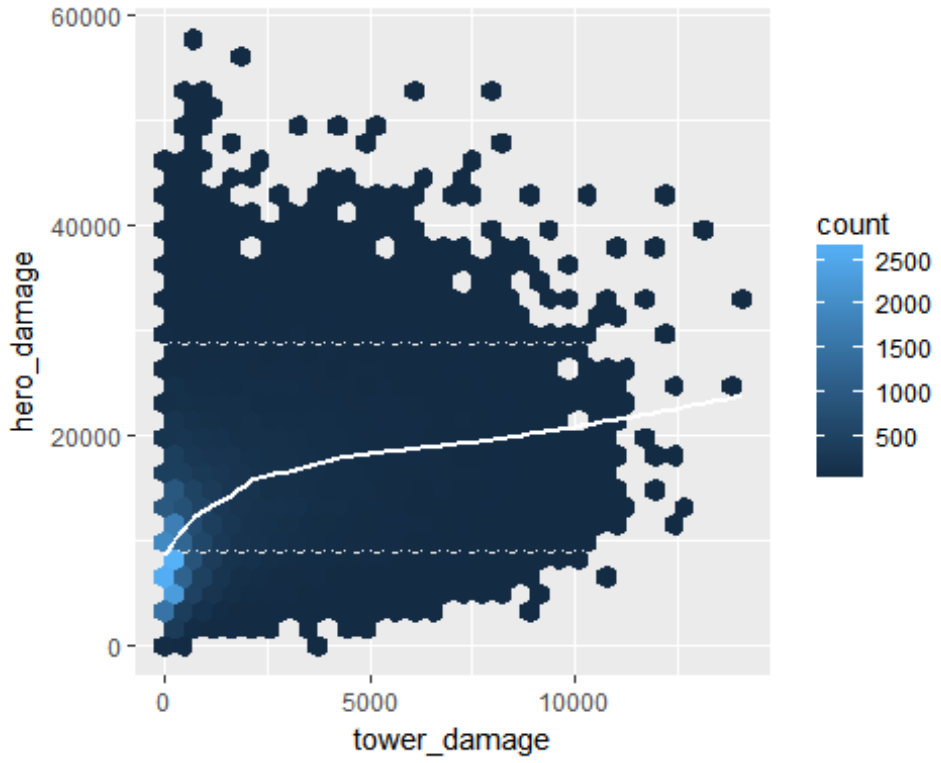
```
## [1] 0.4459638
```

Amount of tower damage a player causes is positively correlated to Amount of hero damage a player causes .which means that a player who damages towers more, may harm heroes more.

5-

```
ggplot(dota2, aes(x=tower_damage, y=hero_damage)) + geom_hex() + geom_smooth(
col= "white" , se =F)
```

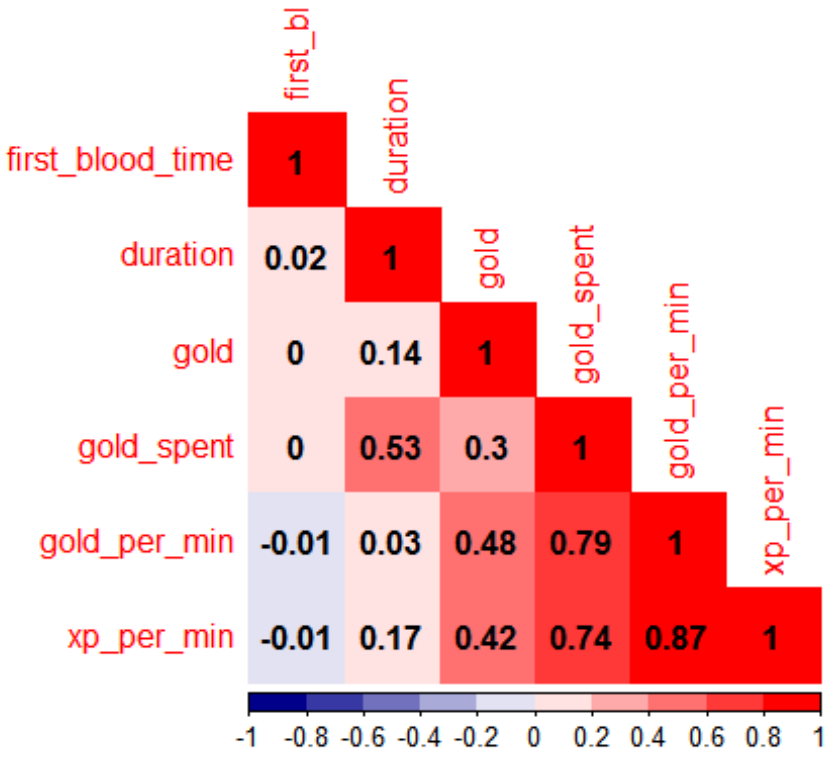
```
## `geom_smooth()` using method = 'gam'
```



For these two variables, number of data points in each bin is almost the same, however, the mode of distribution is around tower_damage = 300, hero_damage = 7000 also , as it can be seen from curve, variables are positively correlated. If we consider big bin size, then many data points may be located in a bin, so we miss distribution details. and if we consider small bin size then some bins might become empty so the histogram doesn't make show the distribution properly.

6-

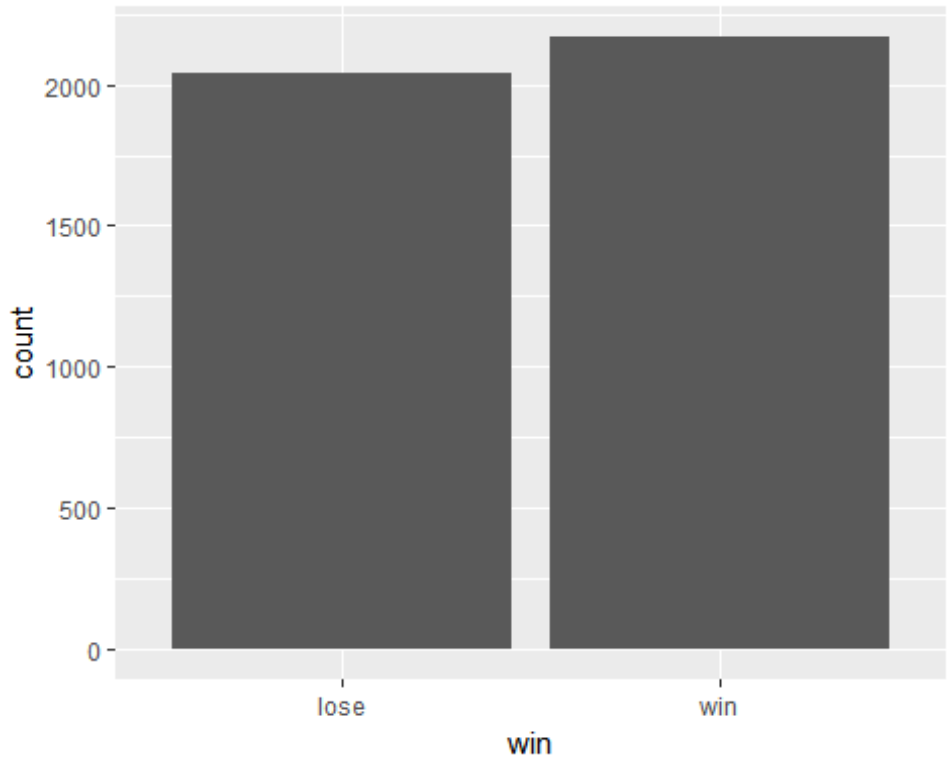
```
six_var = cor(dota2[,c(4,9,16,17,18,19)])
col <- colorRampPalette(c("darkblue", "white", "red"))(10)
corrplot(six_var, method = "color",
         type = "lower", order = "hclust",
         addCoef.col = "black", # Add coefficient of correlation
         col = col
         )
```



Q3-

1-

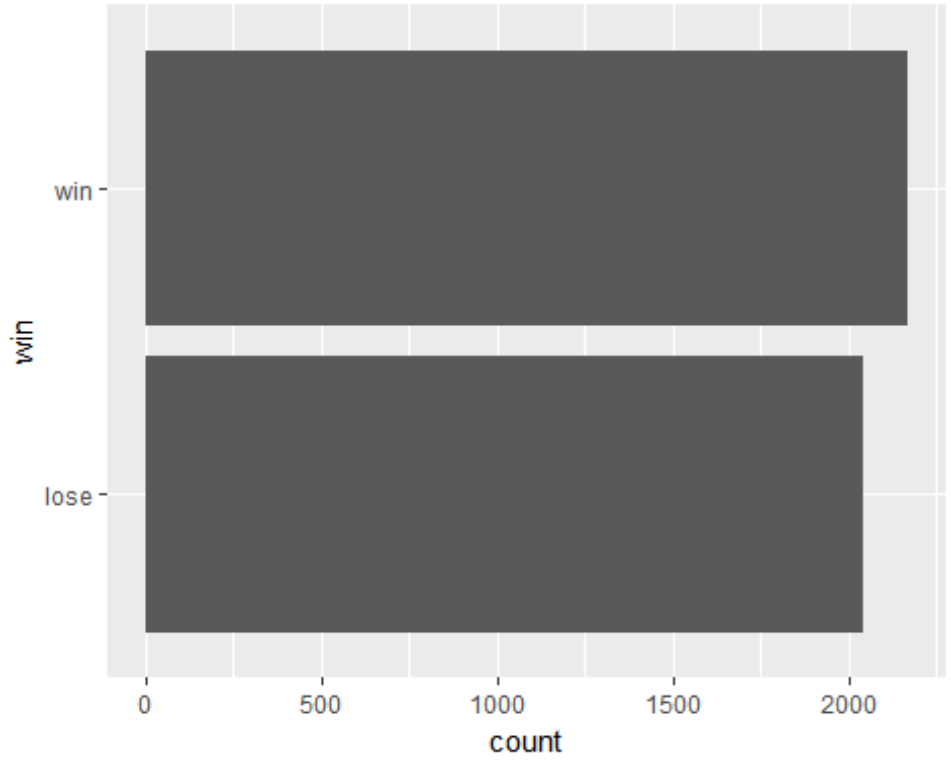
```
u_dota2=dota2[!duplicated(dota2[,'match_id']),]
win <- ifelse(u_dota2$radiant_win==TRUE, c("win"), c("lose"))
ggplot(u_dota2, aes(x=win)) + geom_bar()
```



In this plot, number of Radiant team' victories are plotted against it' defeat (or Dire team victories).It appears that Radiant team was slightly more successful than Dire team.

2-

```
radiant_win <- factor(win, levels=names(sort(table(win))))
ggplot(u_dota2, aes(x=win)) + geom_bar() + coord_flip()
```



3-

```
t = table(win)
t

## win
## lose  win
## 2040 2169
```

Q4-

1-

```
table(dota2$team,dota2$Name)

##
##     Abaddon Alchemist Ancient Apparition Anti-Mage Axe Bane Batrider
##   D     137       389                313       401 192  121       49
##   R     155       409                283       368 191  109       42
##
##     Beastmaster Bloodseeker Bounty Hunter Brewmaster Bristleback
##   D          54         117           278         48         162
##   R          59         142           301         44         188
##
##     Broodmother Centaur Warrunner Chaos Knight Chen Clinkz Clockwerk
##   D          67                70           92   24    106       172
##   R          61                81          113   23    132       199
##
```

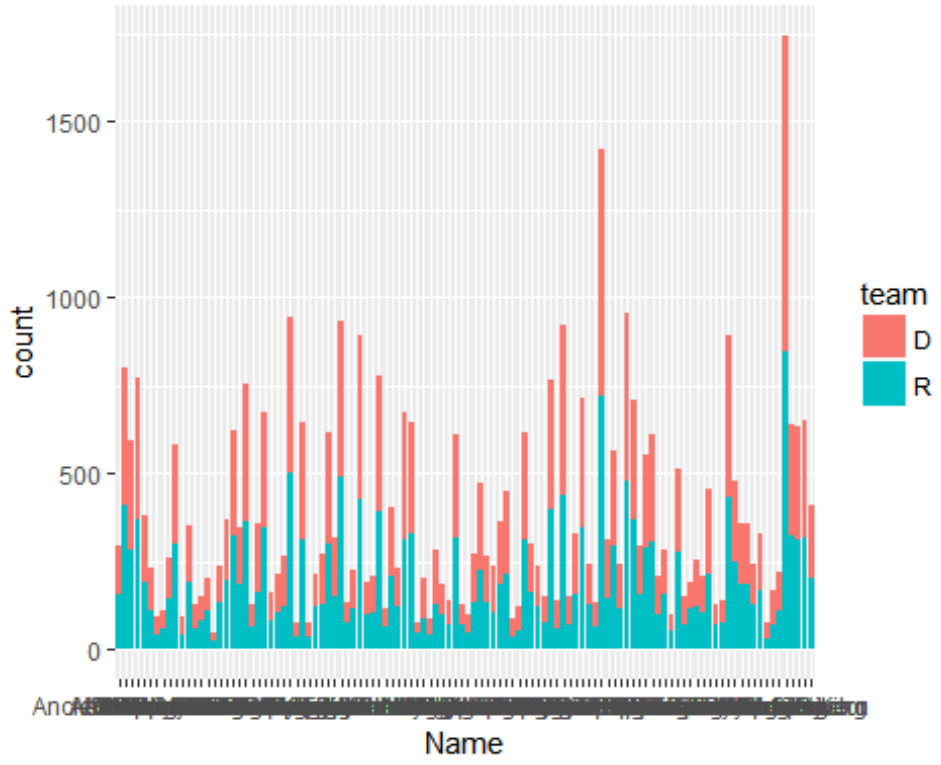
```
##     Crystal Maiden Dark Seer Dazzle Death Prophet Disruptor Doom
##   D            300       158    395            61       196  330
##   R            321       186    361            66       163  345
##
##     Dragon Knight Drow Ranger Earth Spirit Earthshaker Elder Titan
##   D            85         110          146         444          35
##   R            79         106          122         499          38
##
##     Ember Spirit Enchantress Enigma Faceless Void Gyrocopter Huskar
##   D          335          39     96           145        318    167
##   R          310          38    119           128        298    153
##
##     Invoker  Io Jakiro Juggernaut Keeper of the Light Kunkka
##   D     446  59    112        468                  94    105
##   R     488  74    114        425                  98    102
##
##     Legion Commander Leshrac Lich Lifestealer Lina Lion Lone Druid Luna
##   D              385      53  193         109  363  320         31  119
##   R              393      62  209         121  310  327         46   85
##
##     Lycan Magnus Medusa Meepo Mirana Morphling Naga Siren Nature's Prophet
##   D    45    154     86    67    297        61         50              135
##   R    41    129     97    70    315        68         47              135
##
##     Necrophos Night Stalker Nyx Assassin Ogre Magi Omniknight Oracle
##   D       247           136          136       180        239     53
##   R       228           132          103       185        211     35
##
##     Outworld Devourer Phantom Assassin Phantom Lancer Phoenix Puck Pudge
##   D                67              302            137     116   79   370
##   R                54              312            163     120   73   395
##
##     Pugna Queen of Pain Razor Riki Rubick Sand King Shadow Demon
##   D    76           482    78  174    371       119           69
##   R    61           437    71  157    344       126           67
##
##     Shadow Fiend Shadow Shaman Silencer Skywrath Mage Slardar Slark Sniper
##   D          698           165      274           125     479   342    141
##   R          722           146      292           118     476   369    154
##
##     Spectre Spirit Breaker Storm Spirit Sven Techies Templar Assassin
##   D     264            306          110  126      43              235
##   R     289            305           97  158      53              277
##
##     Terrorblade Tidehunter Timbersaw Tinker Tiny Treant Protector
##   D          77         77       131    102  246              58
##   R          71        115       122    104  212              72
##
##     Troll Warlord Tusk Undying Ursa Vengeful Spirit Venomancer Viper
##   D            65  460     229  177             173        118   159
```

```
##    R            73  434     250  183             185        127   169
##
##      Visage Warlock Weaver Windranger Winter Wyvern Witch Doctor
##    D     43     101    111         899            319           319
##    R     31      68    108         845            323           314
##
##      Wraith King Zeus
##    D         333  205
##    R         319  202
```

Value of each cell indicates the number of hero types selected in Radiant or Dire team. As it's expected, number of times a specific hero selected for both teams are nearly the same.

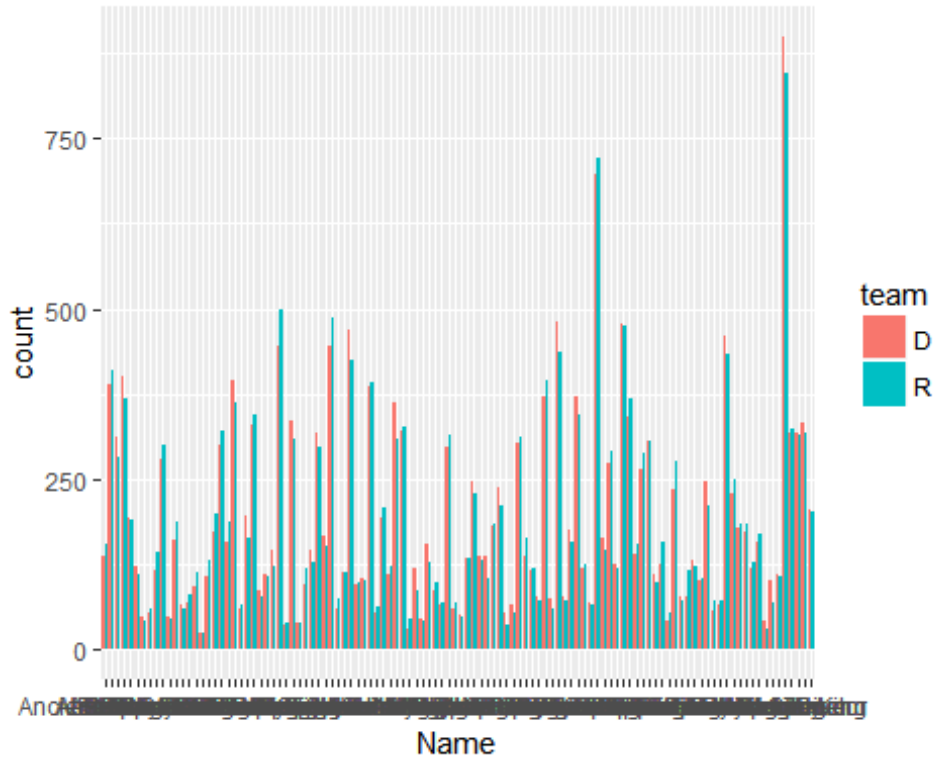
2-

```
ggplot(dota2, aes(Name)) + geom_bar(aes(fill=team))
```



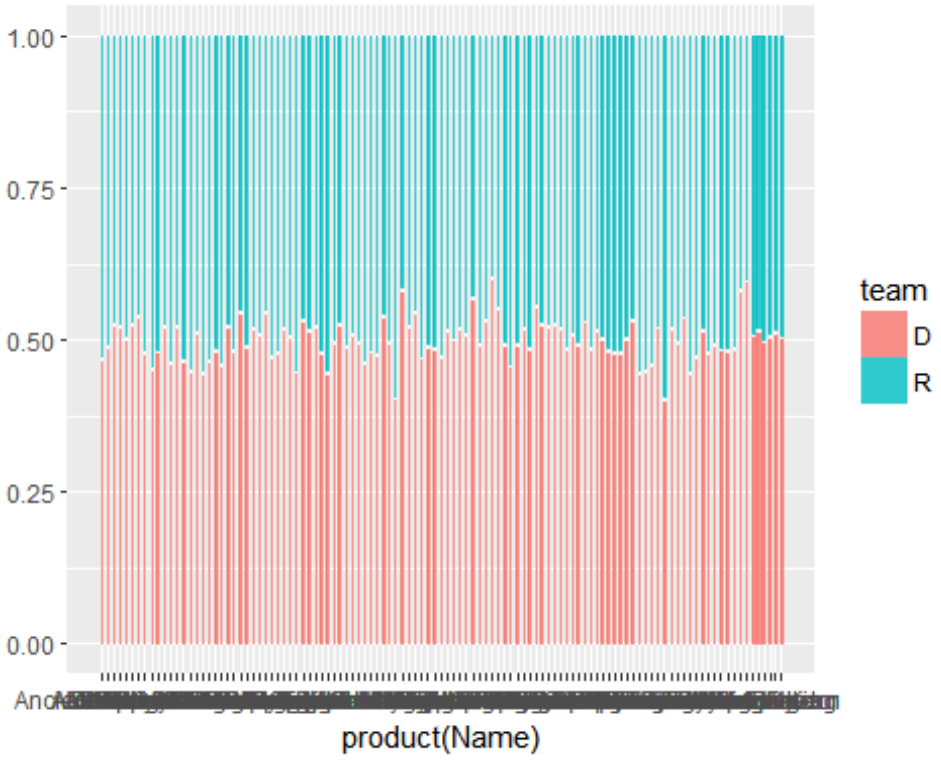
3-

```
ggplot(dota2, aes(Name)) + geom_bar(aes(fill=team),position = "dodge")
```



4-

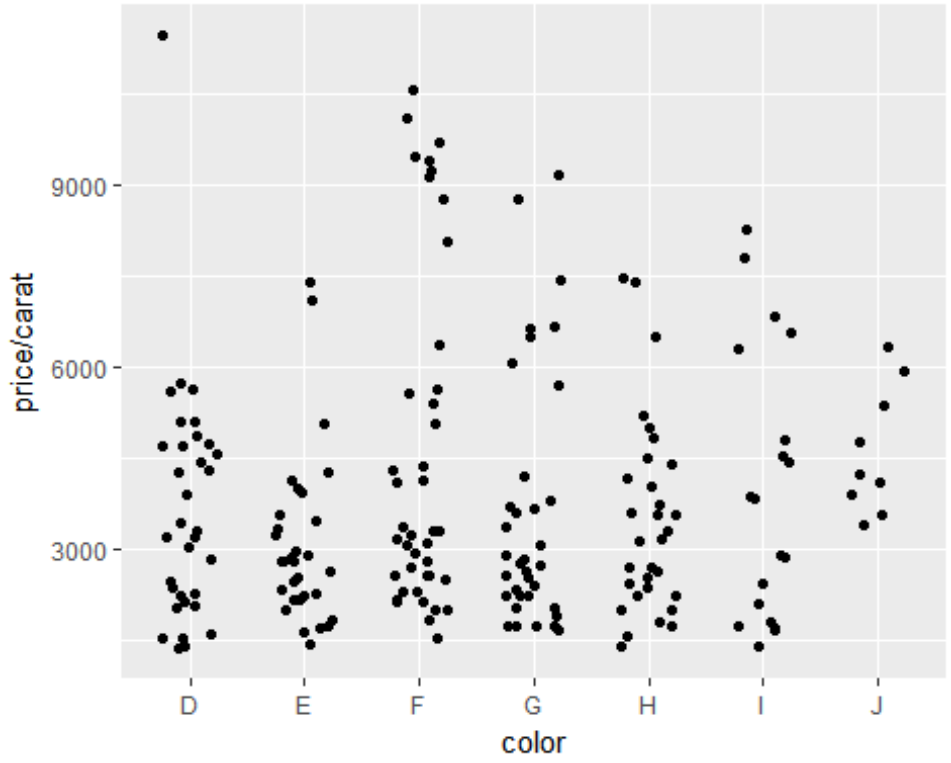
```
ggplot(dota2) + geom_mosaic(aes(x=product(Name) , fill=team))
```



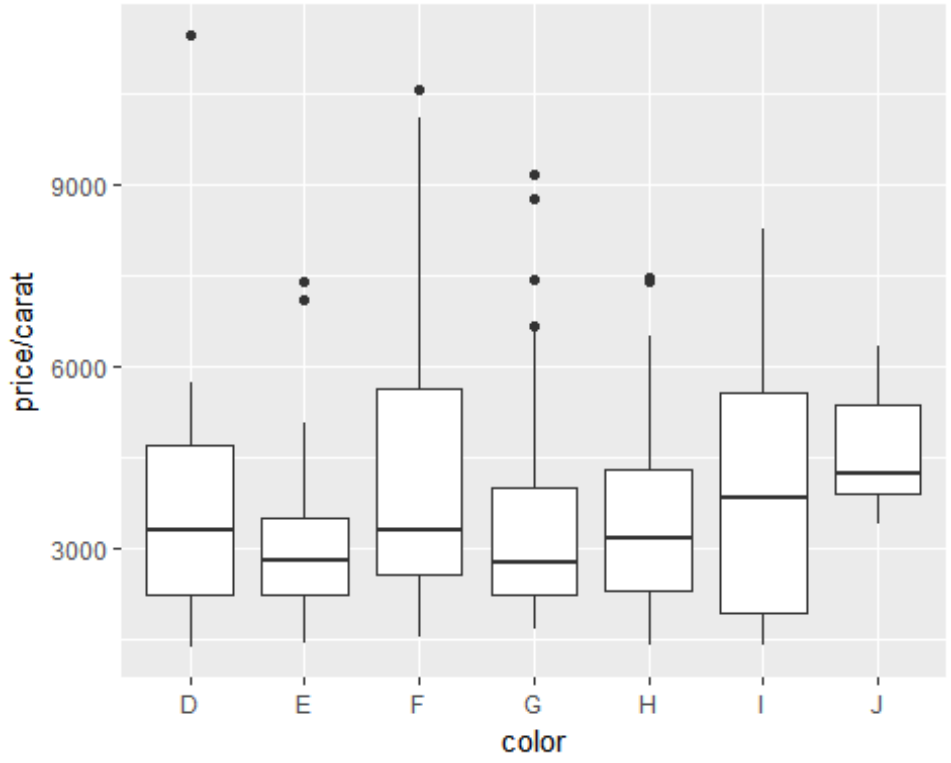
Q5-

1-

```
data("diamonds")
sampled_diamonds = diamonds[sample(nrow(diamonds),200),]
ggplot(sampled_diamonds, aes(color, price/carat)) + geom_jitter(width = 0.25)
```




```
ggplot(sampled_diamonds, aes(color, price/carat)) + geom_boxplot()
```



2-

jitter plot:

Strengths: Jitter plot helps us inspect individual data points also it lets us count number of points in a specific category.

weakness: As noise is added to data points, jitter plot may be interpreted wrongly. Also the plot doesn't visualize any data's statistics.

box plot:

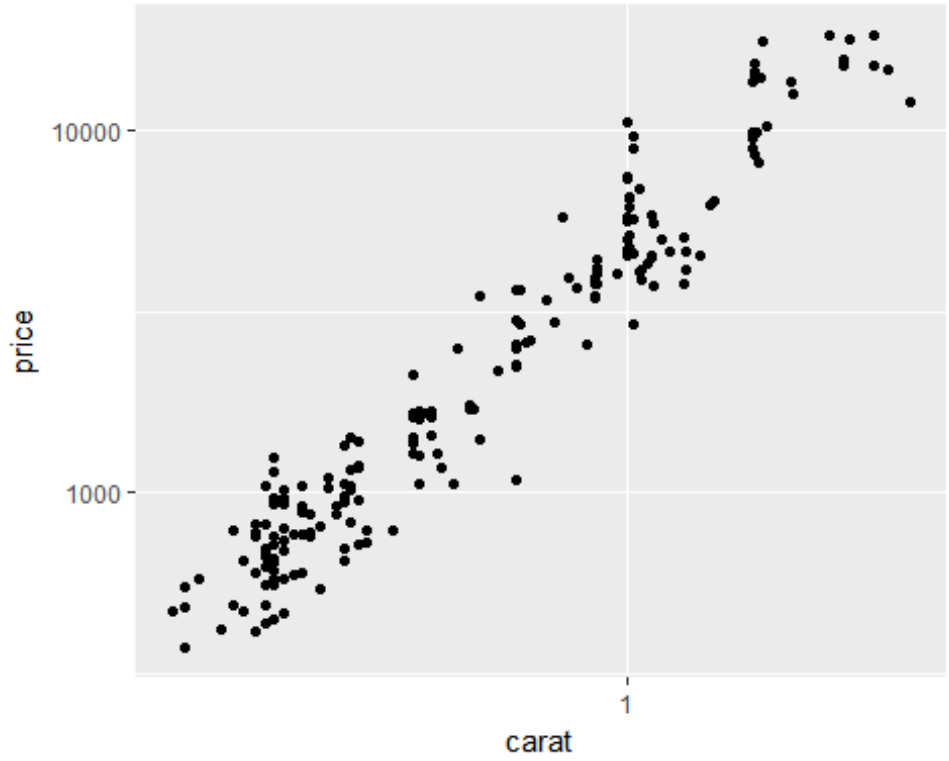
Strengths: The plot visualizes many statistics like quartiles, median, outliers and etc.

weakness: we can't make any decision about individual data points. Also we can't count the number of points in a specific location.

3- From D color to E prices almost decreases. Also we can observe that pure colors have more price outliers in comparison to impure one.

4-

```
ggplot(sampled_diamonds, aes(carat, price)) + geom_point()+ scale_x_continuou
s(trans='log10')+scale_y_continuous(trans='log10')
```



As log transformed carat increases, log transformed price increases almost linearly.

Q6-

1-

```
SE = sd(duration)/length(duration)
lb = mean(duration) - qnorm(0.99)*SE
ub = mean(duration) + qnorm(0.99)*SE
print(sprintf("%#.2f CI = (%#.2f   %#.2f)",0.98,lb,ub))

## [1] "0.98 CI = (2480.88   2481.56)"
```

2- we are 0.98 confident that match's duration is in above interval.

3- H_0 : μ= 3600 (average match duration is one hour)

H_a : μ < 3600 (average match duration is less than one hour)

Conditions:

a. we can assume that match samples are random and independent. (4209 is surely less than 10% of all possible matches)

b. sample size is 4209 so it's greater than 30 and no skewed sample

```
mu = 3600
x_bar = mean(duration)
SE = sd(duration)/sqrt(length(duration))
```

```
Z = (x_bar - mu)/SE
p_value = 2*pnorm(Z)
p_value
```

```
## [1] 0
```

p-value is approximately zero so it's less than 0.05 so we reject H0 which means there is no strong evidence to show that mean is equal to 3600.

4- if we want to test that average match duration is 3500 or less than that, then type two error will be: P($\bar{x}$> 3600 - qnorm(0.99)*SE | μ = 3500)

```
mu = 3500
x_bar = 3600 - qnorm(0.99)*SE
Z = (x_bar - mu)/SE
type2_error = pnorm(Z,lower.tail = FALSE)
type2_error
```

```
## [1] 6.502639e-17
```

type2_error for this test is nearly zero.

5- As type2_error is nearly zero, power of this test is nearly equal to one.it means that if we consider that the average match duration is 3500 then approximately we reject H_0 hypothesis every time.

Q7-

1- we compare average gold spent for different hero types.

H_0: $\mu_1 = \mu_2 = \cdots = \mu_k$

H_a: At least on pair of means are different from each other.

Conditions:

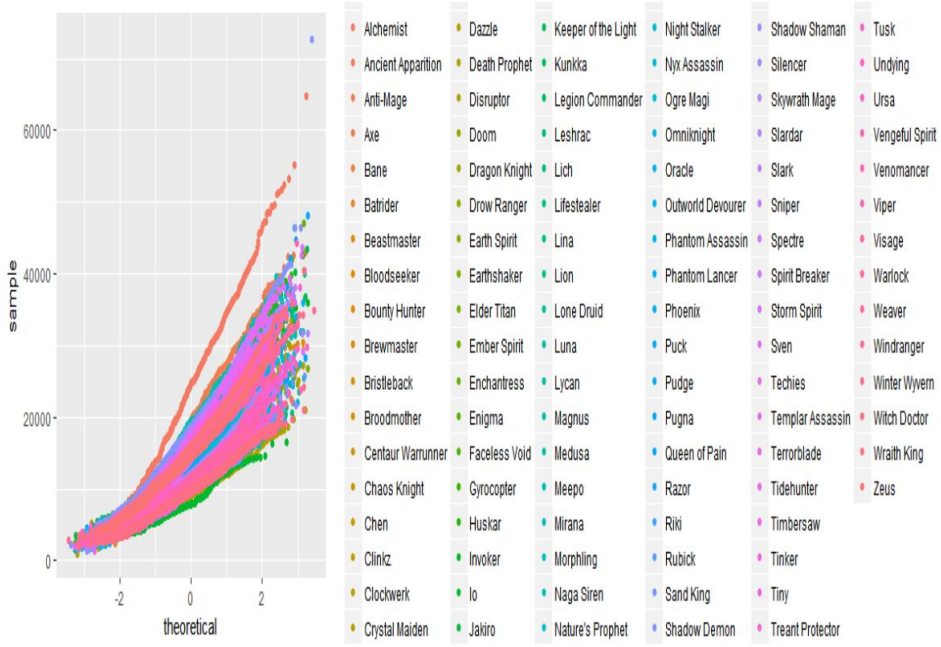
Independence:

within: samples are random and each n_j is less than 10% of respective population

between: hero types are independent from each other so the amount gold spent by them is also independent.

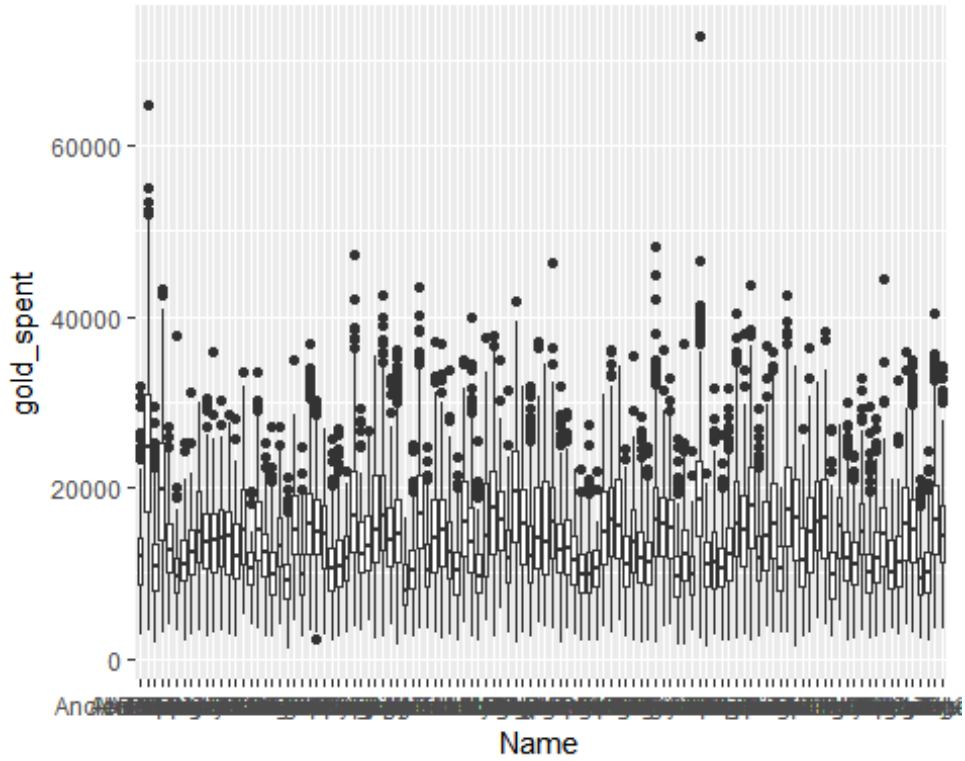
Approximately Normal: Distribution of response variable within each group is approximately normal

```
ggplot(dota2) + stat_qq(aes(sample = gold_spent, colour = factor(Name)))
```



constant variance: variability is constant across different groups.

```
ggplot(dota2, aes(Name, gold_spent)) + geom_boxplot()
```




```
# dota_splited = split(dota2,dota2$Name)

total_mu = mean(dota2$gold_spent)
```

```r
M=dota2[,mean(gold_spent),by=sort(Name)]$V1
df_T = nrow(dota2) - 1
df_G = length(M) - 1
df_E = df_T - df_G
cnt = dota2[,.N,by=sort(Name)]$N
SST = sum((dota2$gold_spent - total_mu)^2)
SSG = sum(((M - total_mu)^2)*cnt)
SSE = SST - SSG
MSG = SSG / df_G
MSE = SSE / df_E
f = MSG/MSE
pf (f, df_G , df_E,lower.tail = FALSE)
```

```
## [1] 1.24968e-19
```

2- p-value is nearly zero and less than 0.05. so we reject H0 hypothesis which means that, there's no strong evidence to show that the average gold spent by different hero types are the same.