

Star classification

Vitalii Morskyi – 166731 | P4 | 2FS-DI

06 czerwca 2022

Opis wybranych danych do analizy

Wybrany został zbiór danych dla widmowej klasyfikacji gwiazd, który jest wynikiem badania Sloan Digital Sky Survey Data Release 17. Zawiera on dane potrzebne dla klasyfikacji gwiazd, galaktyk i kwazarów. Ułatwiona postać zestawu danych została zamieszczona na stronie Kaggle, natomiast oryginał można pobrać na stronie SDSS. Opis wyników badania można znaleźć na stronie SDSS.

W astronomii widmowa klasyfikacja gwiazd to klasyfikacja na podstawie ich charakterystyk spektralnych. Schemat klasyfikacji galaktyk, kwazarów i gwiazd jest jednym z najbardziej fundamentalnych w astronomii. Skatalogowanie gwiazd i ich rozmieszczenie na niebie doprowadziło do zrozumienia, że tworzą one naszą własną galaktykę a, gdy zrozumieliśmy, że Andromeda jest odrębną galaktyką od naszej, zaczęliśmy badać inne galaktyki i budować potężniejsze teleskopy.

Opis poszczególnych kolumn

Zestaw danych zawiera 100 000 wierszy, w każdym 17 cech, które opisują jedną galaktykę, gwiazdę lub kwazar, a mianowicie:

- **obj_ID** – Identyfikator obiektu, unikalna wartość, która identyfikuje obiekt w katalogu obrazów używanym przez CAS.
- **alpha** – Kąt rektascensji (w epoce J2000).
- **delta** – Kąt deklinacji (w epoce J2000).
- **u** – Intensywność promieniowania widma ultrafioletowego.
- **g** – Intensywność promieniowania widma zielonego.
- **r** – Intensywność promieniowania widma czerwonego.
- **i** – Intensywność promieniowania widma bliskiej podczerwieni.
- **z** – Intensywność promieniowania widma podczerwieni.
- **run_ID** – Numer przebiegu używany do identyfikacji konkretnego skanu.
- **rerun_ID** – Numer ponownego przebiegu dla określenia sposobu przetwarzania obrazu.
- **cam_col** – Kolumna kamery do identyfikacji linii skanowania w przebiegu.
- **field_ID** – Numer identyfikacyjny pola.
- **spec_obj_ID** – Unikalny identyfikator używany dla optycznych obiektów spektroskopowych (oznacza to, że 2 różne obserwacje o tym samym **spec_obj_ID** muszą dzielić klasę wyjściową).
- **class** – Klasa obiektu (galaktyka, gwiazda lub kwazar).
- **rsershift** – Wartość przesunięcia ku czerwieni na podstawie wzrostu długości fali.
- **plate** – Numer identyfikacyjny płyty, identyfikuje każdą płytę w SDSS.
- **MJD** – Zmodyfikowana data juliańska, używana do wskazania, kiedy dany fragment danych SDSS został pobrany.
- **fiber_ID** – Numer identyfikacyjny włókna, które skierowało światło na płaszczyznę ogniskowania.

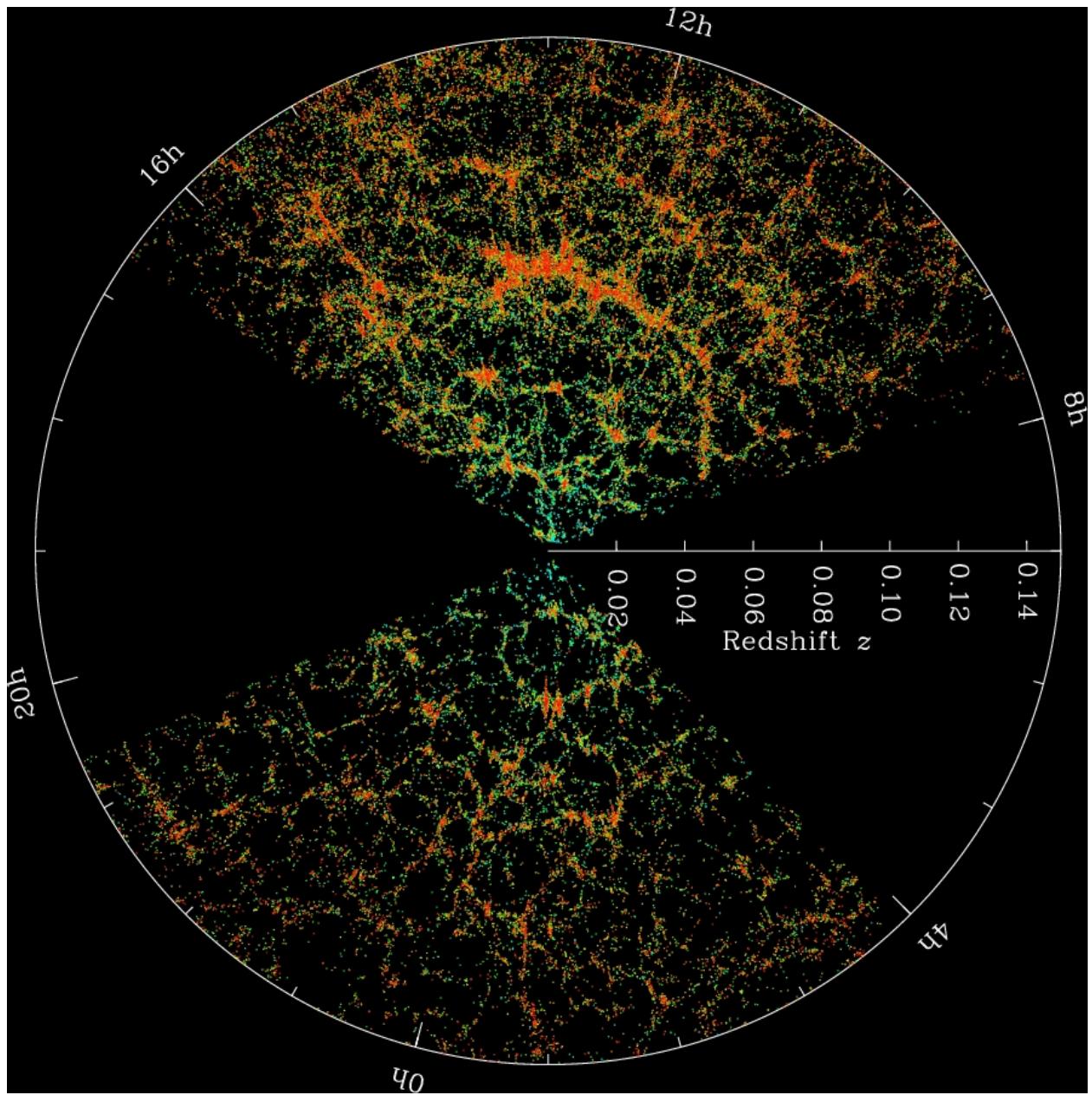


Figure 1: Mapa SDSS Wszechświata. Każda kropka to galaktyka; kolor pokazuje gęstość lokalną (źródło).

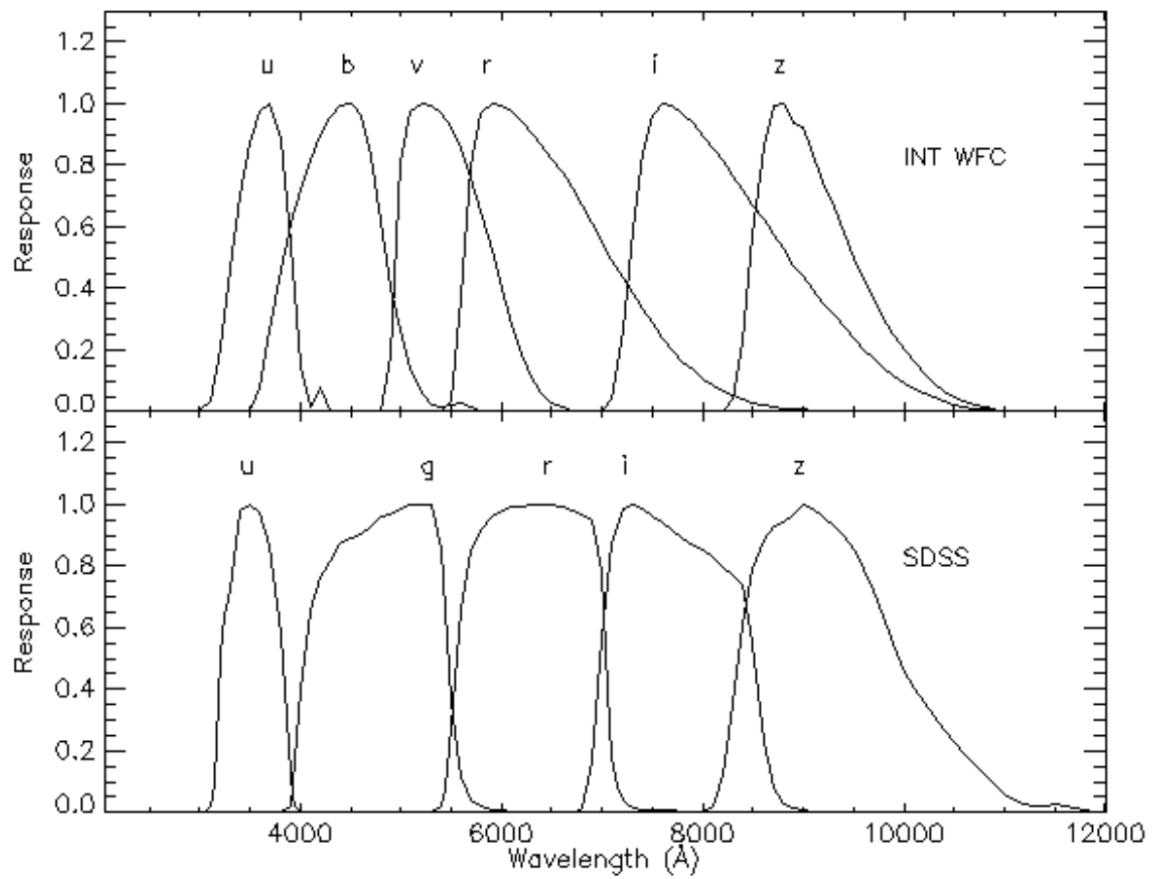


Figure 2: Porównanie obecnych standardowych filtrów INT WFC z zestawem SDSS (źródło).

Wczytywanie i filtracja danych

Wczytujemy ramkę danych z pliku CSV (Comma Separated Values), używając funkcji `read.csv`. Używając indeksowania ramki danych, usuwamy stąd zbędne dla analizy kolumny oraz 79544 wiersz, w którym dane są zepsute. Funkcja `colnames` wypisuje listę nagłówków ramki danych. W kolumnie `class` zmieniamy typ danych z `character` na `factor`, co ułatwia w przyszłości interakcję z ramką danych.

```
full_df <- read.csv("star-classification-data.csv")
df <- full_df[-79544, !colnames(full_df) %in%
  c("obj_ID", "rerun_ID", "run_ID", "cam_col", "field_ID", "spec_obj_ID", "plate", "MJD", "fiber_ID")]
df$class <- factor(df$class, labels = c("Galaxy", "Quasar", "Star"))
head(df)

##      alpha      delta      u      g      r      i      z   class
## 1 135.6891 32.4946318 23.87882 22.27530 20.39501 19.16573 18.79371 Galaxy
## 2 144.8261 31.2741849 24.77759 22.83188 22.58444 21.16812 21.61427 Galaxy
## 3 142.1888 35.5824442 25.26307 22.66389 20.60976 19.34857 18.94827 Galaxy
## 4 338.7410 -0.4028276 22.13682 23.77656 21.61162 20.50454 19.25010 Galaxy
## 5 345.2826 21.1838656 19.43718 17.58028 16.49747 15.97711 15.54461 Galaxy
## 6 340.9951 20.5894763 23.48827 23.33776 21.32195 20.25615 19.54544 Quasar
##      redshift
## 1 0.6347936
## 2 0.7791360
## 3 0.6441945
## 4 0.9323456
## 5 0.1161227
## 6 1.4246590
```

Aby lepiej zrozumieć, z jakimi danymi mamy do czynienia, wypisujemy podsumowanie ramki danych za pomocą polecenia `summary`.

```
summary(df)

##      alpha      delta      u      g
##  Min.   : 0.0055   Min.   :-18.785   Min.   :11.00   Min.   :10.50
##  1st Qu.:127.5177  1st Qu.: 5.147   1st Qu.:20.35   1st Qu.:18.97
##  Median :180.9005  Median :23.646   Median :22.18   Median :21.10
##  Mean   :177.6287  Mean   :24.136   Mean   :22.08   Mean   :20.63
##  3rd Qu.:233.8950  3rd Qu.:39.902   3rd Qu.:23.69   3rd Qu.:22.12
##  Max.   :359.9998  Max.   :83.001   Max.   :32.78   Max.   :31.60
##      r          i          z   class
##  Min.   : 9.822   Min.   : 9.47   Min.   : 9.612   Galaxy:59445
##  1st Qu.:18.136   1st Qu.:17.73   1st Qu.:17.461   Quasar:18961
##  Median :20.125   Median :19.41   Median :19.005   Star  :21593
##  Mean   :19.646   Mean   :19.08   Mean   :18.769
##  3rd Qu.:21.045   3rd Qu.:20.40   3rd Qu.:19.921
##  Max.   :29.572   Max.   :32.14   Max.   :29.384
##      redshift
##  Min.   :-0.009971
##  1st Qu.: 0.054522
##  Median : 0.424176
##  Mean   : 0.576667
##  3rd Qu.: 0.704172
##  Max.   : 7.011245
```

Dla dalszej analizy zestawu danych wczytujemy następujące paczki:

- **tidyverse** – Ugruntowany zbiór pakietów R zaprojektowanych do analizy danych. W jego zestaw wchodzą takie paczki, jak:
 - **ggplot2** – system do deklaratywnego tworzenia grafiki, oparty na “*The Grammar of Graphics*”;
 - **dplyr** – zapewnia gramatykę manipulacji danymi, zapewniając spójny zestaw poleceń, które rozwiązują najczęstsze problemy związane z manipulacją danymi;
 - Inne, niewykorzystane w tym projekcie.
- **moments** – Zapewnia funkcje do wyliczenia momentów, kumulacji, skośności, kurtozy i powiązanych testów.
- **ggridges** – Zapewnia funkcji do rysowania grzbietowych wykresów gęstości.
- **viridis** – Dostarcza serię układów kolorów.
- **hrbrthemes** – Kompilacja dodatkowych motywów, skal i narzędzi ggplot2.
- **GGally** – Rozszerzenie do ggplot2.

```
library(tidyverse)
library(moments)
library(ggridges)
library(viridis)
library(GGally)
```

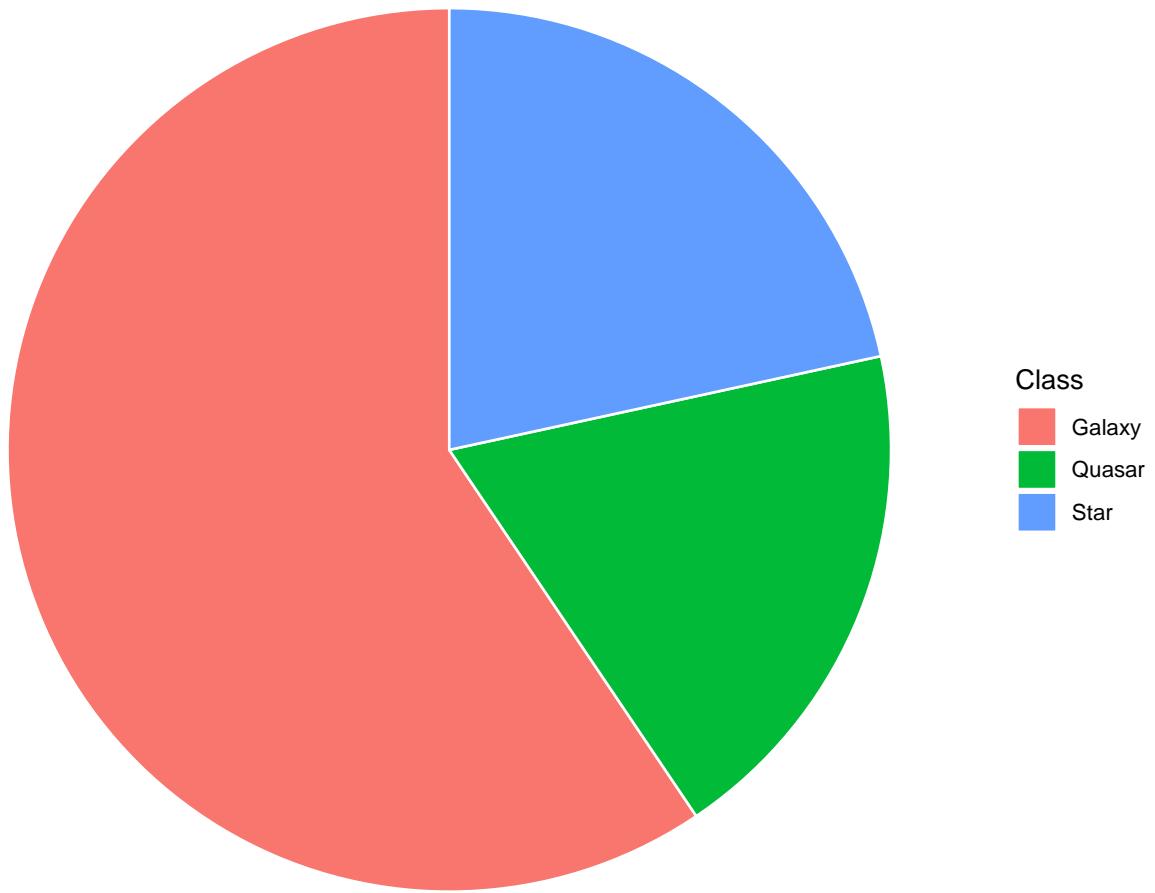
Przed rozpoczęciem analizy spójrzmy na rozkład gwiazd, galaktyk i kwazarów w zestawie danych. W tym celu używamy funkcji `group_by` dla rozdzielenia ramki danych według odpowiednich obiektów, liczymy liczbę elementów każdego obiektu za pomocą funkcji `count` i rysujemy wykres kołowy używając funkcji `geom_bar` w połączeniu z funkcją `coord_polar` z pakietu `ggplot2`. Pierwsza tworzy ułożony w stos wykres słupkowy z pojedynczym słupkiem, a druga – zmienia typ współrzędnych na biegunowe, tym samym przekształcając wykres w kołowy.

Dla ulepszenia wyglądu wykresu używamy dużej liczby funkcji z pakietu `ggplot2`, takich jak `labs` – zamiana nagłówków i `theme_grey` – ustawienie motywu kolorystycznego wykresu. W kodzie używamy również operatora `%>%` z pakietu `magrittr`, który pozwala tworzyć pipeline w języku R. Pakiet `magrittr` nie był wczytywany jawnie, ponieważ jest on domyślnie wymagany przez pakiet `dplyr`.

Z wykresu widzimy, że w zbiorze danych jest najwięcej galaktyk, dalej idą kwazary i gwiazdy.

```
df %>%
  group_by(class) %>%
  count() %>%
  ggplot(aes(x = "", y = n, fill = class)) +
  geom_bar(stat = "identity", width = 1, color = "white") +
  coord_polar("y", start = 0) +
  labs(title = "Pie Chart of the class distribution", fill = "Class") +
  theme_void()
```

Pie Chart of the class distribution



Wyznaczenie podstawowych parametrów opisowych

Postanowiono obliczyć następujące parametry:

- **Średnia arytmetyczna, harmoniczna i geometryczna**
- **Kwantyle** rzędu $\frac{1}{4}$, $\frac{2}{4}$ (medianą) i $\frac{3}{4}$
- **Dominanta**
- **Wariancja i odchylenie standardowe**
- **Odchylenie przeciętne** (od mediany i od średniej) – miara zmienności próby, która jest bardziej odporna na wartości odstające w zestawie danych niż odchylenie standardowe. Co więcej odchylenie przeciętne od mediany, działa lepiej niż standardowe odchylenie z rozkładami bez średniej lub wariancji, takimi jak rozkład Cauchy'ego.
- **Współczynnik zmienności** – miara zmienności próby, która jest niezależna od jednostki pomiaru, więc jest liczbą bezwymiarową. Pozwala to na porównania zbiorów danych z różnymi jednostkami

pomiarowymi lub bardzo różnymi średnimi. Z innej strony, gdy średnia wartość jest bliska zeru, współczynnik zmienności zbliża się do nieskończoności i dlatego jest wrażliwy na małe wartości średniej.

- **Rozstęp**
- **Odchylenie ćwiartkowe**
- **Współczynnik i wskaźnik asymetrii** – miara asymetrii rozkładu, która określa w którą stronę rozkład jest bardziej “nachylony”.
- **Współczynnik spłaszczenia**
- **Moment zwykły 1 i 2 rzędu** – 1 rzędu = średnia.
- **Moment centralny 1, 2, 3 i 4 rzędu** – 2 rzędu = wariancja, $3 = \text{współczynnik asymetrii} \times \sigma^3$, $4 = \text{współczynnik spłaszczenia} \times \sigma^4$.
- **Moment centralny absolutny 1 i 2 rzędu**

Do obliczenia niektórych powyższych parametrów użyto już gotowych funkcji, takich jak `mean`, `sd`, `var`, `quantile`, `range`, `skewness`, `kurtosis` i `all.moments` (ostatnie 3 z pakietu `moments`). Inne parametry wyznaczono używając własnych funkcji.

```
#' Calculate harmonic mean
mean.harmonic <- function(series) {
  return(length(series) / sum(1 / series))
}

#' Calculate geometric mean
mean.geometric <- function(series) {
  return(prod(series)^(1 / length(series)))
}

#' Calculate mode (dominant value)
mode.stat <- function(series) {
  ux <- unique(series)
  tab <- tabulate(match(series, ux))
  return(ux[tab == max(tab)])
}

#' Calculate absolute average deviation from other value
average.deviation <- function(series, from_value) {
  return(sum(abs(series - from_value)) / length(series))
}

#' Calculate coefficient of variation
coefficient.of.variation <- function(series) {
  return(sd(series) / mean(series))
}

#' Calculate quantile deviation
quantile.deviation <- function(series) {
  return((quantile(series, 0.75) - quantile(series, 0.25)) / 2)
}

#' Calculate asymmetry coefficient
asymmetry.coefficient <- function(series) {
  return(all.moments(series, central = TRUE, absolute = FALSE, order.max = 3)[4] / (sd(series)^3))
}

#' Calculate a lot of parameters of the series
describe <- function(series) {
```

```

srednia.arytmetyczna <- mean(series)
srednia.harmoniczna <- mean.harmonic(series)
srednia.geometryczna <- mean.geometric(series)
kwantyle <- quantile(series, c(0.25, 0.5, 0.75))
dominanta <- mean(mode.stat(series))
odchylenie.przecietne.mediania <- average.deviation(series, mean(series))
odchylenie.przecietne.srednia <- average.deviation(series, median(series))
wariancja <- var(series)
odchylenie.standardowe <- sd(series)
wspolczynnik.zmienności <- coefficient.of.variation(series)
odchylenie.cwiartkowe <- quantile.deviation(series)
rozstep <- diff(range(series))
wspolczynnik.asymetrii <- skewness(series)
wskaznik.asymetrii <- asymmetry.coefficient(series)
wspolczynnik.spłaszczenia <- kurtosis(series)
momenty <- all.moments(series, central = FALSE, absolute = FALSE, order.max = 3)
momenty.centralne <- all.moments(series, central = TRUE, absolute = FALSE, order.max = 5)
momenty.centralne.absolutne <- all.moments(series, central = TRUE, absolute = TRUE, order.max = 3)
data.frame(
  Statystyka = c("Średnia arytmetyczna", "Średnia harmoniczna", "Średnia geometryczna",
    "Kwantyl rzędu 1/4", "Kwantyl rzędu 2/4 (mediania)", "Kwartyl rzędu 3/4",
    "Dominanta", "Odchylenie przeciętne od mediany",
    "Odchylenie przeciętne od średniej", "Wariancja", "Odchylenie standardowe",
    "Współczynnik zmienności", "Odchylenie cwiartkowe", "Rozstęp",
    "Współczynnik asymetrii", "Wskaźnik asymetrii", "Współczynnik spłaszczenia",
    "Moment zwykły 1 rzędu", "Moment zwykły 2 rzędu",
    "Moment centralny 1 rzędu", "Moment centralny 2 rzędu", "Moment centralny 3 rzędu",
    "Moment centralny 4 rzędu",
    "Moment centralny absolutny 1 rzędu", "Moment centralny absolutny 2 rzędu"
  ),
  Wartosc = c(srednia.arytmetyczna, srednia.harmoniczna, srednia.geometryczna,
    kwantyle[1], kwantyle[2], kwantyle[3],
    dominanta, odchylenie.przecietne.mediania,
    odchylenie.przecietne.srednia, wariancja, odchylenie.standardowe,
    wspolczynnik.zmienności, odchylenie.cwiartkowe, rozstep,
    wspolczynnik.asymetrii, wskaznik.asymetrii, wspolczynnik.spłaszczenia,
    momenty[2], momenty[3],
    momenty.centralne[2], momenty.centralne[3], momenty.centralne[4], momenty.centralne[5],
    momenty.centralne.absolutne[2], momenty.centralne.absolutne[3]
  )
)
)
}

describe(df[df$class == "Quasar", "u"])

##                               Statystyka      Wartosc
## 1             Średnia arytmetyczna 2.154762e+01
## 2             Średnia harmoniczna 2.144534e+01
## 3             Średnia geometryczna        Inf
## 4             Kwantyl rzędu 1/4 2.063764e+01
## 5             Kwantyl rzędu 2/4 (mediania) 2.150324e+01
## 6             Kwartyl rzędu 3/4 2.228647e+01
## 7             Dominanta 2.463466e+01
## 8 Odchylenie przeciętne od mediany 1.117679e+00

```

```

## 9 Odchylenie przeciętne od średniej 1.117076e+00
## 10 Wariancja 2.237655e+00
## 11 Odchylenie standardowe 1.495879e+00
## 12 Współczynnik zmienności 6.942202e-02
## 13 Odchylenie ćwiartkowe 8.244150e-01
## 14 Rozstęp 2.178516e+01
## 15 Współczynnik asymetrii 4.324960e-01
## 16 Wskaźnik asymetrii 4.324618e-01
## 17 Współczynnik spłaszczenia 4.208909e+00
## 18 Moment zwykły 1 rzędu 2.154762e+01
## 19 Moment zwykły 2 rzędu 4.665374e+02
## 20 Moment centralny 1 rzędu -6.304048e-16
## 21 Moment centralny 2 rzędu 2.237537e+00
## 22 Moment centralny 3 rzędu 1.447563e+00
## 23 Moment centralny 4 rzędu 2.107221e+01
## 24 Moment centralny absolutny 1 rzędu 1.117679e+00
## 25 Moment centralny absolutny 2 rzędu 2.237537e+00

```

Graficzna prezentacja danych

W kolumnach `alpha`, `delta` podano współrzędne wszystkich obserwowanych obiektów w układzie równikowym równonocnym, a w kolumnie `redshift` – wzgledna odległość obiektu od Ziemi. Używając kolumn `alpha` i `redshift` możemy więc względnie pokazać jak rozłożone są analizowane obiekty w odniesieniu do Ziemi. Zauważmy, że na wykresie jest podana tylko 2D projekcja układu trójwymiarowego, więc jest to tylko i wyłącznie wykres pozorny.

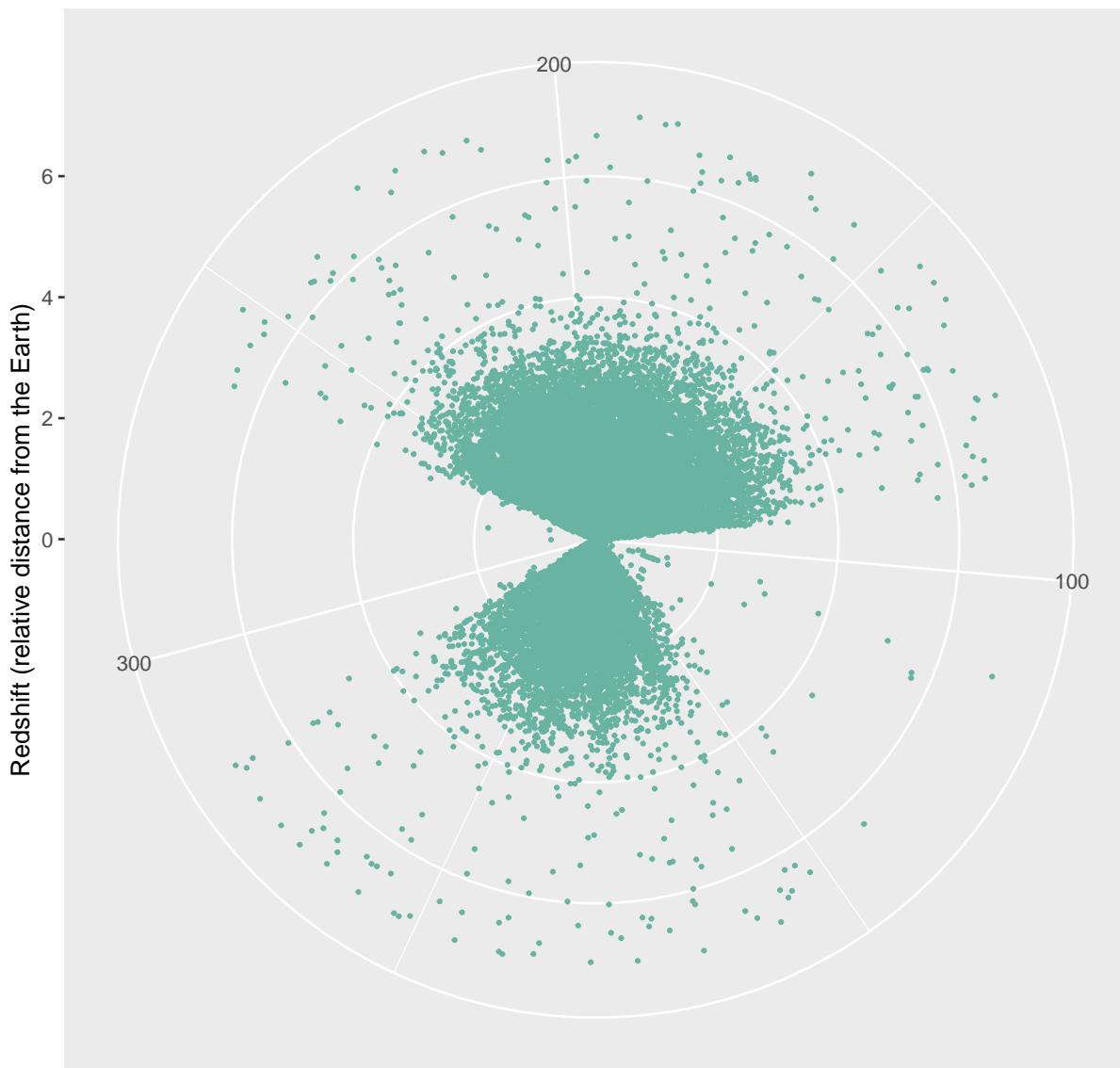
Widzimy, że wykres zgadza się ze zdjęciem, które jest na oficjalnej stronie SDSS.

```

ggplot(df, aes(x = `alpha`, y = `redshift`)) +
  geom_point(col = "#69b3a2", size = 0.5) +
  coord_polar("x", start = 11 * pi / 12, direction=-1) +
  labs(title = "Part of the SDSS map of the Universe", x="", y="Redshift (relative distance from the Earth")
  theme_grey()

```

Part of the SDSS map of the Universe



Skorelowane cechy

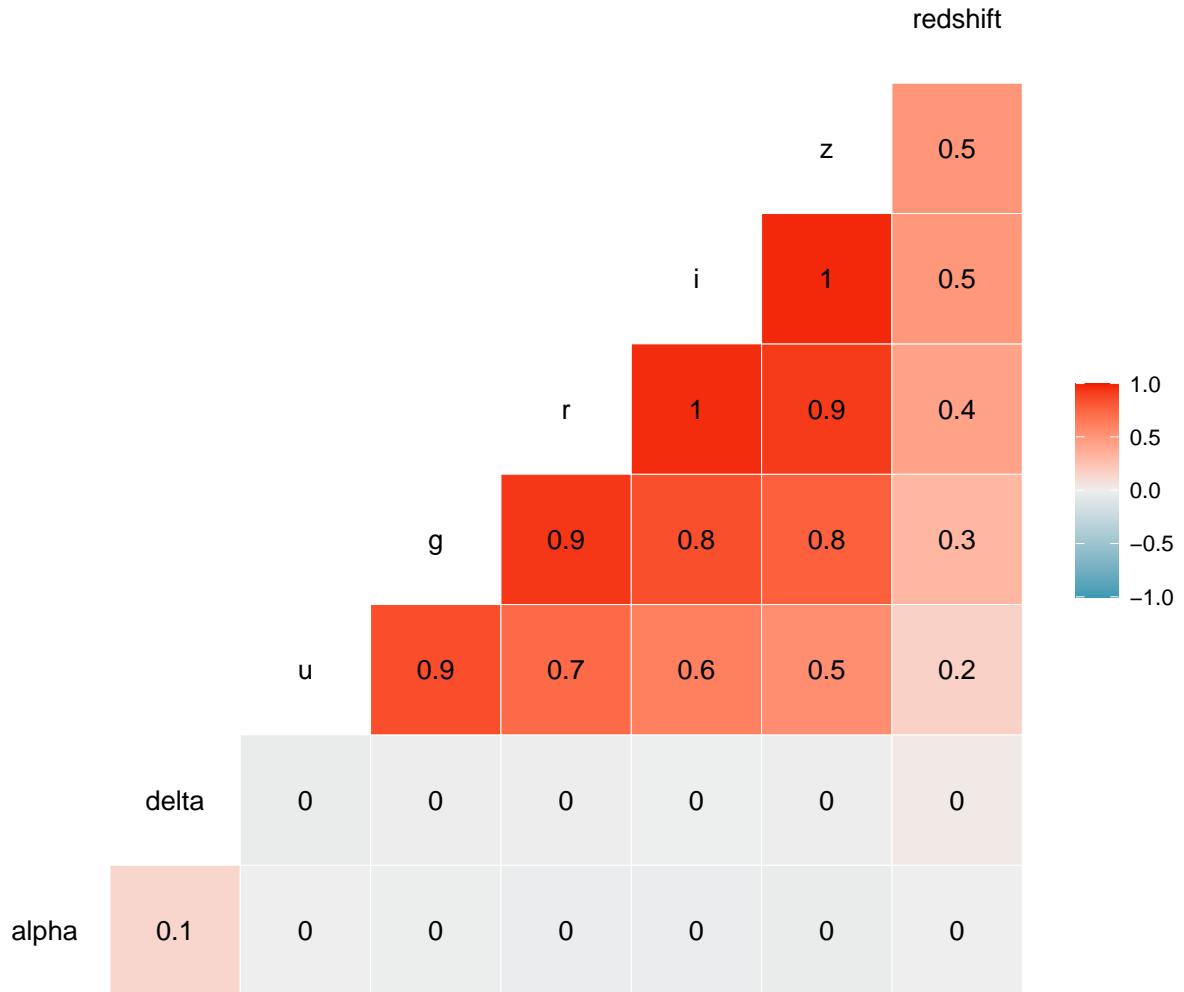
Na początek chcielibyśmy wiedzieć które kolumny są między sobą skorelowane. W tym celu rysujemy mapę ciepła korelacji Pearsona wszystkich badanych kolumn. Oczekujemy zobaczyć dużą korelację pomiędzy intensywnością promieniowania różnych widm, natomiast korelacja pomiędzy innymi cechami zestawu danych nie powinna występować.

W celu narysowania wykresu używamy funkcji `ggcorr` z pakietu `GGally`. Jak widzimy, oczekiwania się sprawdziły. Dodatni znak współczynników korelacji pomiędzy wartościami `u`, `g`, `r`, `i`, `z` wskazuje na to, że występują wprost proporcjonalne zależności pomiędzy tymi wartościami. Widzimy również słabą korelację pomiędzy kolumnami `redshift` i `z`, ponieważ cecha `redshift` zależy od intensywności promieniowania widma podczerwieni.

```
ggcorr(df[c("alpha", "delta", "u", "g", "r", "i", "z", "redshift")],  
       method = c("everything", "pearson"),
```

```
    label = TRUE
```

```
)
```



Zbadamy teraz istotność współczynnika korelacji na poziomie istotności $\alpha = 0.001$. Testujemy hipotezę $H_0 : \rho = 0$ przy pomocy funkcji testowej

$$t = \frac{R}{\sqrt{1 - R^2}} \sqrt{n - 2}$$

która ma rozkład t-Studenta z $(n-2)$ stopniami swobody (*przypuszczamy, że $(U, REDSHIFT)$ ma rozkład dwuwymiarowy normalny*).

```
Rval <- cor(df$redshift, df$u)
Tres <- Rval / sqrt(1 - Rval ^ 2) * sqrt(dim(df)[1] - 2)
tk <- qt(1 - 0.001 / 2, dim(df)[1] - 2)
cat("t =", Tres)
```

```

## t = 53.50072
cat("\nCritical interval K: (-inf, ", -tk, "] U [", tk, ", inf)", sep = "")
```

```

##
```

```

## Critical interval K: (-inf, -3.290624] U [3.290624, inf)
```

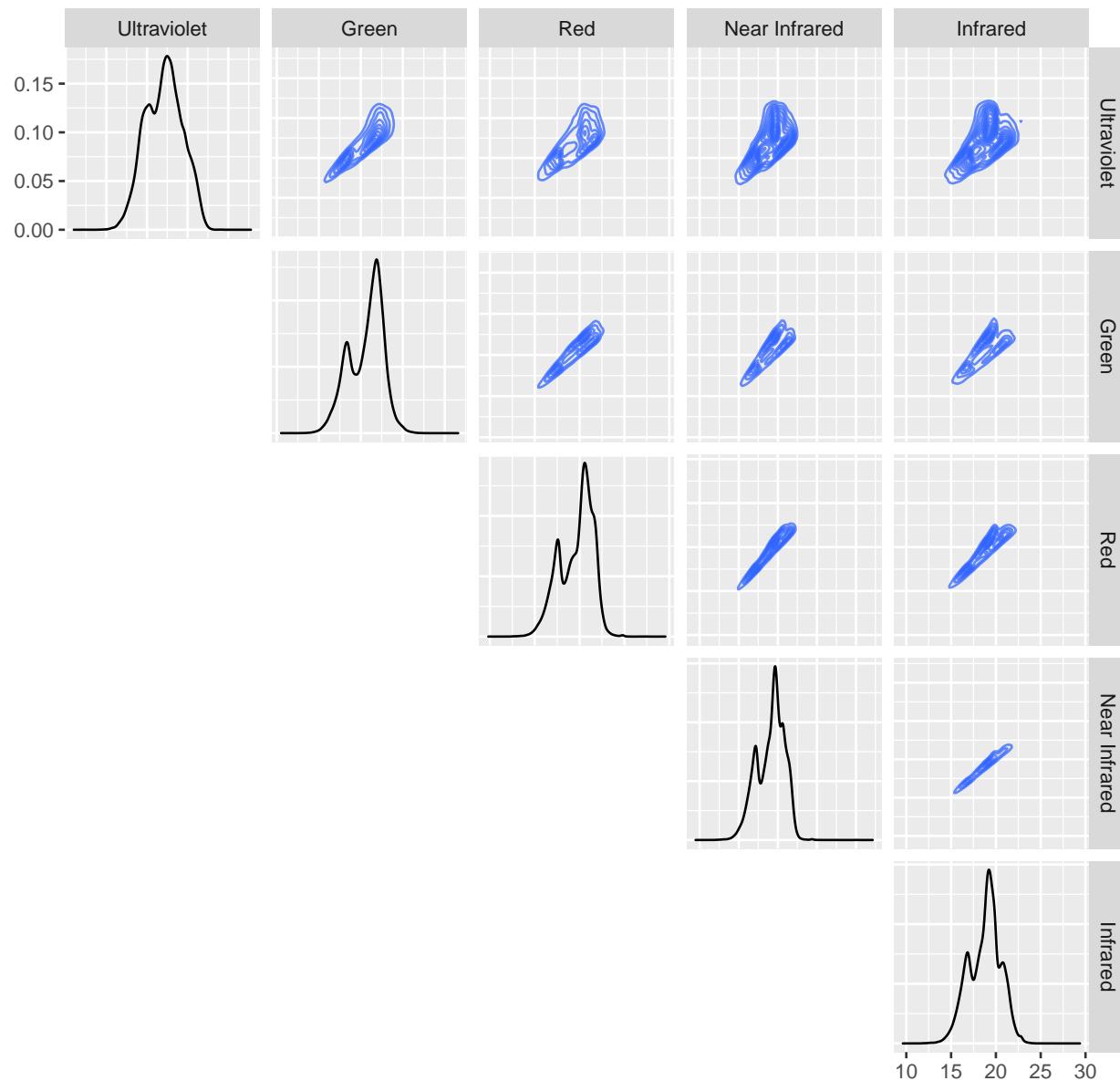
Ponieważ wartość testowa mieści się w przedziale krytycznym, to należy odrzucić hipotezę H_0 . Innymi słowy, oznacza to, że współczynnik korelacji pomiędzy cechami u , $\text{redshift } \rho$ z populacji nie jest równy 0. Patrząc na powyższy wykres, możemy stwierdzić, że jeżeli współczynnik korelacji nie jest zerowy nawet dla najmniej skorelowanych cech, to na raczej jest on niezerowy dla bardziej skorelowanych cech również.

Na następnym rysunku podano wykresy dwuwymiarowej gęstości rozkładu cech intensywności promieniowania różnych widm. Widzimy, że zależność jest wprost proporcjonalna, zgodnie z informacją, którą otrzymaliśmy licząc współczynnik korelacji.

```

ggpairs(df[c("u", "g", "r", "i", "z")],
        upper = list(continuous = wrap("density", alpha = 0.75)),
        diag = list(continuous = wrap("densityDiag")),
        lower = "blank",
        title = "2D density distribution of a random variable",
        columnLabels = c("Ultraviolet", "Green", "Red", "Near Infrared", "Infrared"),
        proportions = "auto"
)
```

2D density distribution of a random variable



Na następnym wykresie w bardziej przejrzysty sposób rysujemy zależność pomiędzy intensywnością promieniowania widma ultrafioletowego a podczerwieni. Ponieważ cechy u i z są skorelowane, to rysujemy prostą regresji empirycznej zmiennej losowej z względem zmiennej losowej u .

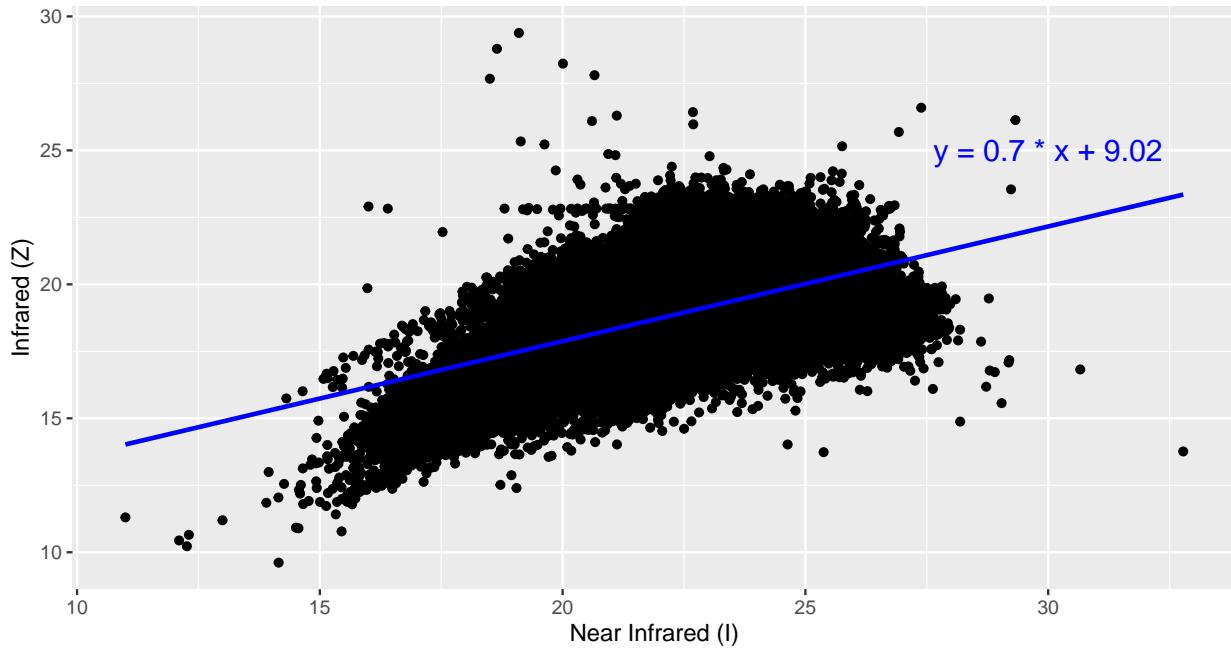
```
ykxb <- lm(df$u ~ df$z)[[1]]
library(extrafont)
font_import()
loadfonts(device = "win")
ggplot(df, aes(x = `u`, y = `z`)) +
  geom_point() +
  geom_smooth(method = "lm", color = "blue", fill = "#69b3a2", se = TRUE, level = 0.999) +
  labs(title = "Simple empirical regression of a random variable Z against I",
       x = "Near Infrared (I)",
       y = "Infrared (Z)") +
```

```

annotate("text", x = 30, y = 25, color = "blue", size = 5,
        label = paste("y =", round(ykxb[2], 2), "* x +", round(ykxb[1], 2))) +
theme_grey()

```

Simple empirical regression of a random variable Z against I



Gęstość rozkładu zmiennej losowej

Narysujmy teraz gęstości intensywności promieniowania każdego widma osobno dla gwiazd, galaktyk i kważarów. W tym celu przygotowujemy ramkę danych `df.wavelengths`, która będzie zawierała 3 cechy: - Klasę obiektu – gwiazda, galaktyka lub kważar; - Widmo promieniowania – ultrafioletowe, zielone, czerwone, bliskiej podczerwieni, podczerwieni; - Intensywność promieniowania – liczba. Chcielibyśmy również porównać całkowitą intensywność promieniowania każdej klasy obiektów, - więc tworzymy nową kolumnę `avg-total` w ramce `df`, - do której zamieszczamy średnie wartości intensywności promieniowania dla każdego badanego obiektu.

```

df.wavelengths <- data.frame(Intensity = double(),
                                Class = factor(),
                                Color = factor())

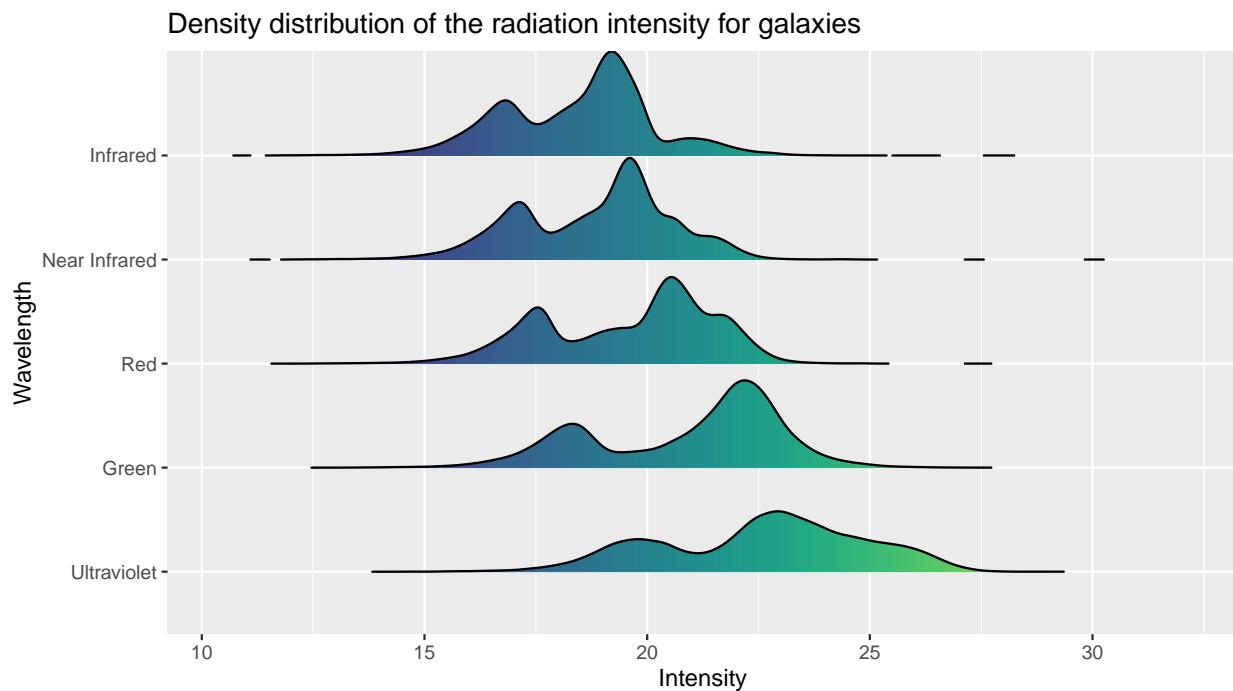
colnames_vec <- list(c("u", "g", "r", "i", "z"), c("Ultraviolet", "Green", "Red", "Near Infrared", "Infrared"))
for (i in seq_along(colnames_vec)[[1]])) {
  df.temp <- df[, c(colnames_vec[[1]][i], "class")]
  df.temp <- cbind(df.temp, factor(colnames_vec[[2]][i]))
  colnames(df.temp) <- c("Intensity", "Class", "Wavelength")
  df.wavelengths <- rbind(df.wavelengths, df.temp)
}
df[, "avg-total"] <- apply(df[, c("u", "g", "r", "i", "z")], 1, mean)
str(df.wavelengths)

## 'data.frame': 499995 obs. of 3 variables:
## $ Intensity : num 23.9 24.8 25.3 22.1 19.4 ...
## $ Class     : Factor w/ 3 levels "Galaxy","Quasar",...: 1 1 1 1 1 2 2 1 1 3 ...
## $ Wavelength: Factor w/ 5 levels "Ultraviolet",...: 1 1 1 1 1 1 1 1 1 ...

```

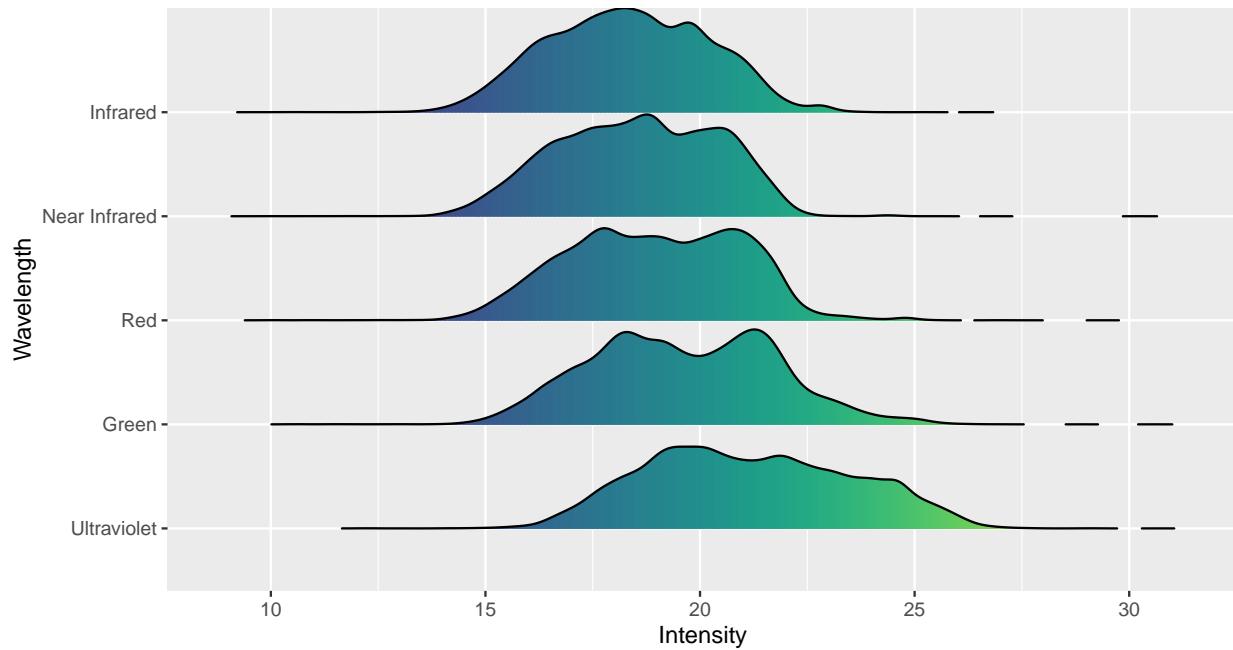
Używając wcześniej przygotowanej ramki `df.wavelengths` rysujemy wykres gęstości rozkładu intensywności promieniowania każdego widma galaktyk. Z otrzymanego wykresu możemy stwierdzić, że średnia obserwowana intensywność promieniowania zwiększa się razem ze zwiększeniem długości fali elektromagnetycznej. Tworząc analogiczne wykresy dla gwiazd i kwazarów, zauważamy podobną zależność.

```
ggplot(subset(df.wavelengths, Class == "Galaxy"), aes(x = `Intensity`, y = `Wavelength`, fill = ...x...)) +
  geom_density_ridges_gradient(scale = 1, rel_min_height = 0.0001) +
  scale_fill_viridis(name = "Intensity", option = "D") +
  labs(title = 'Density distribution of the radiation intensity for galaxies') +
  theme_grey() +
  theme(
    legend.position = "none",
    panel.spacing = unit(0.1, "lines"),
    strip.text.x = element_text(size = 10)
  )
```



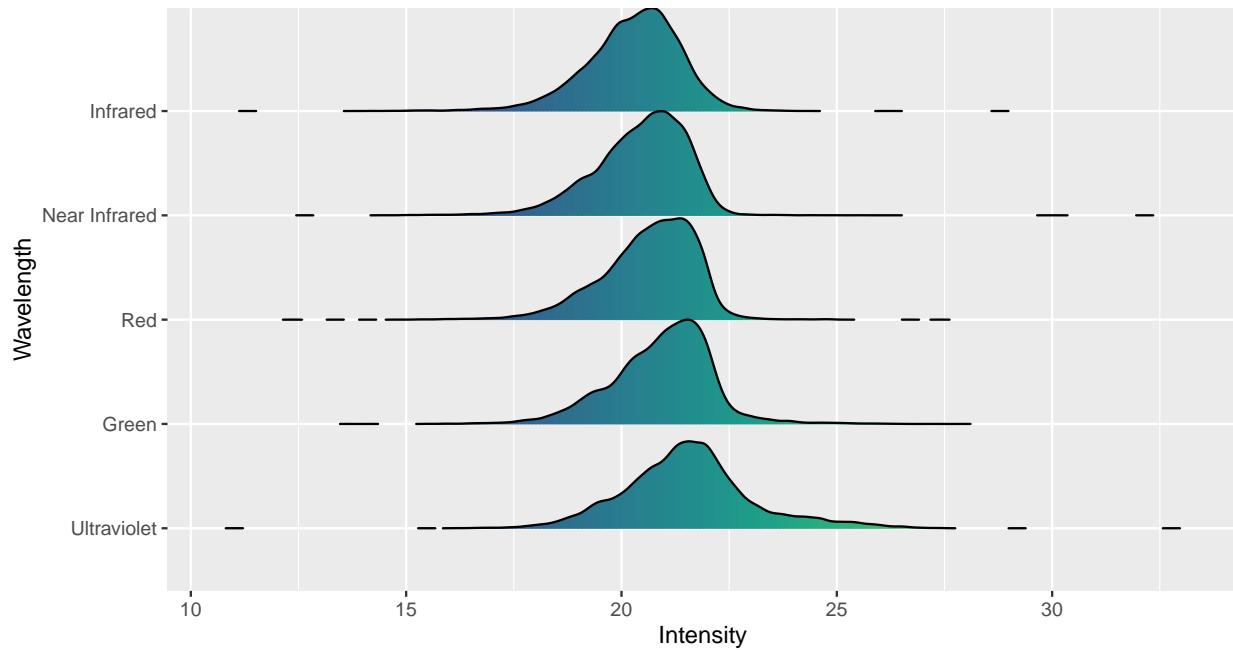
```
ggplot(subset(df.wavelengths, Class == "Star"), aes(x = `Intensity`, y = `Wavelength`, fill = ...x...)) +
  geom_density_ridges_gradient(scale = 1, rel_min_height = 0.0001) +
  scale_fill_viridis(name = "Intensity", option = "D") +
  labs(title = 'Density distribution of the radiation intensity for stars') +
  theme_grey() +
  theme(
    legend.position = "none",
    panel.spacing = unit(0.1, "lines"),
    strip.text.x = element_text(size = 10)
  )
```

Density distribution of the radiation intensity for stars



```
ggplot(subset(df.wavelengths, Class == "Quasar"), aes(x = `Intensity`, y = `Wavelength`, fill = ..x..)) +  
  geom_density_ridges_gradient(scale = 1, rel_min_height = 0.0001) +  
  scale_fill_viridis(name = "Intensity", option = "D") +  
  labs(title = 'Density distribution of the radiation intensity for quasars') +  
  theme_grey() +  
  theme(  
    legend.position = "none",  
    panel.spacing = unit(0.1, "lines"),  
    strip.text.x = element_text(size = 10)  
)
```

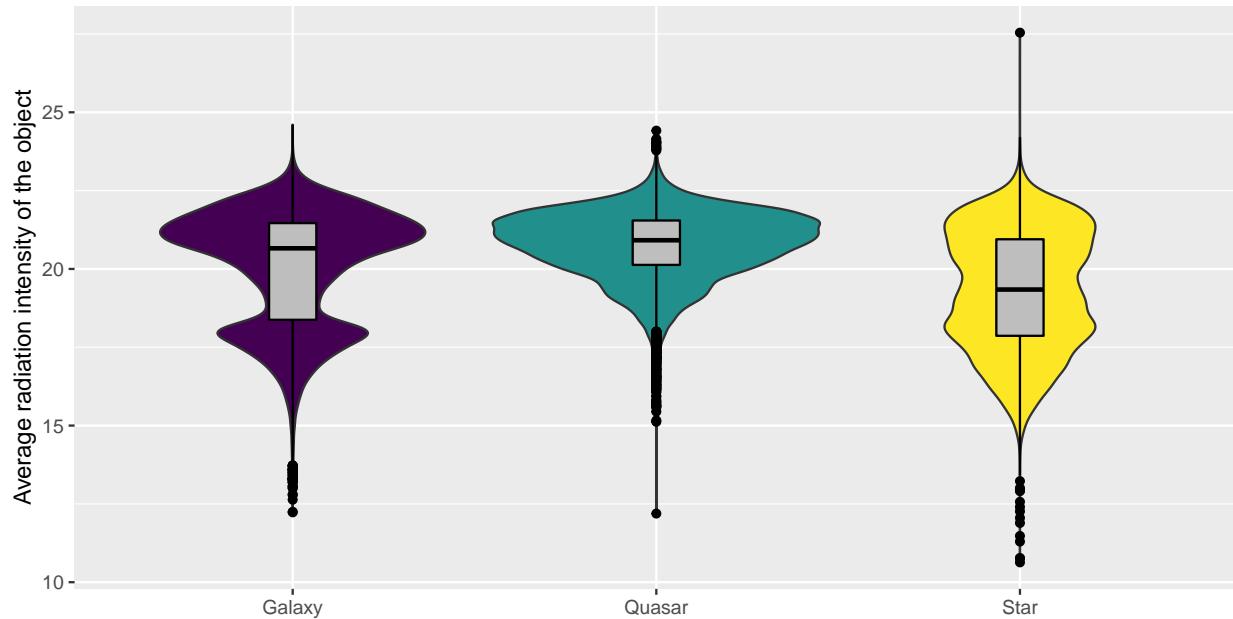
Density distribution of the radiation intensity for quasars



Porównajmy teraz średnią intensywność promieniowania obiektów poszczególnych klas. W tym celu rysujemy wykres skrzypcowy łącznie z wykresem pułapkowym dla każdej klasy obiektów.

```
ggplot(df, aes(x = `class`, y = `avg-total`, fill = `class`)) +
  geom_violin() +
  geom_boxplot(width = 0.13, color = "black", fill = "grey", alpha = 1) +
  scale_fill_viridis(discrete = TRUE) +
  labs(title = "Comparison of the object radiation density distribution per class",
       x = "",
       y = "Average radiation intensity of the object") +
  theme_grey() +
  theme(
    legend.position = "none",
  )
```

Comparison of the object radiation density distribution per class



Z otrzymanego wykresu możemy stwierdzić, że średnie promieniowanie galaktyk i kwazarów jest podobne bliskie siebie, natomiast rozstęp ćwiartkowy dla klasy galaktyk jest bardziej przesunięty do dołu. Chcemy zatem sprawdzić hipotezę H_0 , mówiącą, że średnie promieniowanie galaktyk i kwazarów jest równe. Hipotezę alternatywną H_1 określamy następująco: średnie promieniowanie kwazarów jest jednak większe od średniego promieniowania galaktyk. Przyjmujemy poziom istotności testu $\alpha = 0.005$.

Ponieważ wartość testowa $p < \alpha$, to należy odrzucić hipotezę H_0 na korzyść hipotezy H_1 mówiącej, że średnie promieniowanie kwazarów jest większe od średniego promieniowania galaktyk.

```
t.test(df[df$class == "Quasar", "avg-total"],
       df[df$class == "Galaxy", "avg-total"],
       conf.level = 1 - 0.005,
       alternative = "greater",
       mu = 0
)

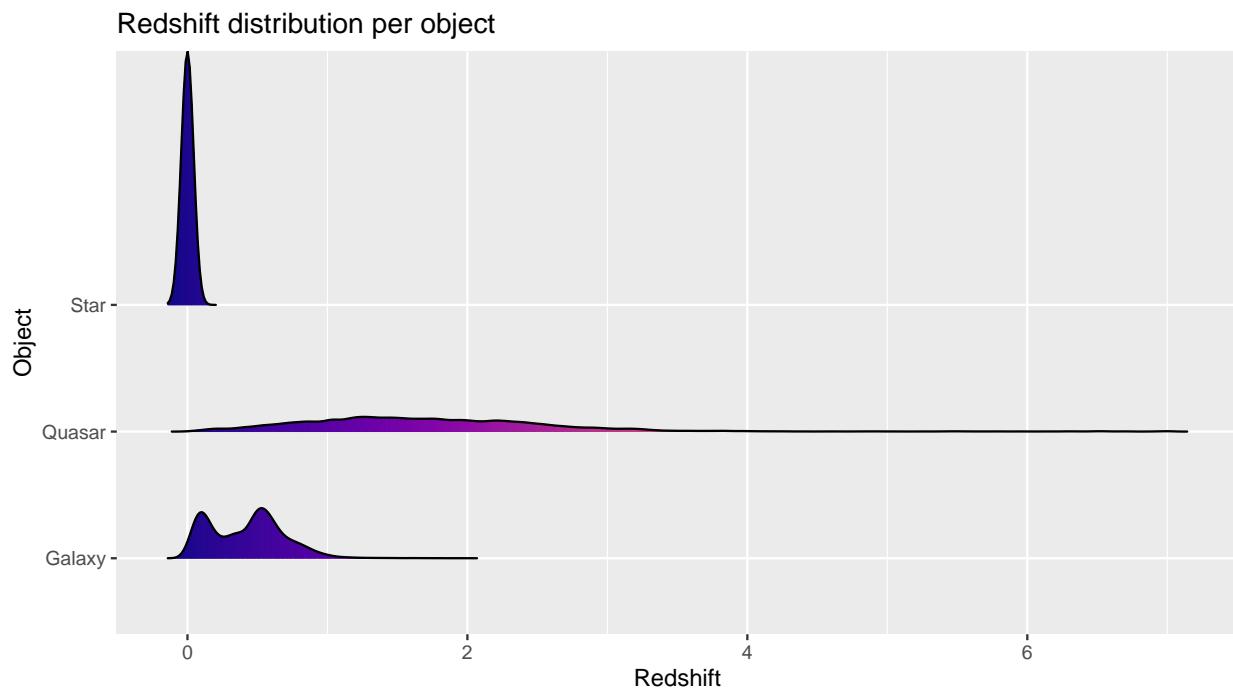
##
##  Welch Two Sample t-test
##
## data: df[df$class == "Quasar", "avg-total"] and df[df$class == "Galaxy", "avg-total"]
## t = 63.064, df = 53931, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 99.5 percent confidence interval:
##  0.6548385      Inf
## sample estimates:
## mean of x mean of y
## 20.75916  20.07644
```

Dystrybuanta empiryczna, szereg rozdzielczy

Narysujemy teraz wykres gęstości rozkładu cechy `redshift` dla każdej klasy badanych obiektów. Wiedząc, że cecha `redshift` odpowiada za odległość badanego obiektu od ziemi, z otrzymanego rysunku możemy stwierdzić, że w zestawie danych opisane tylko te gwiazdy, które znajdują się blisko ziemi. O wiele więcej

informacji mamy o bardziej oddalonych galaktykach i kwazarach. Prawdopodobnie zależność taka występuje z uwagi na rozmiar badanych obiektów: galaktyki i kwazary są czasami w miliardy razy większe od gwiazd i dlatego jest łatwiej ich zaobserwować.

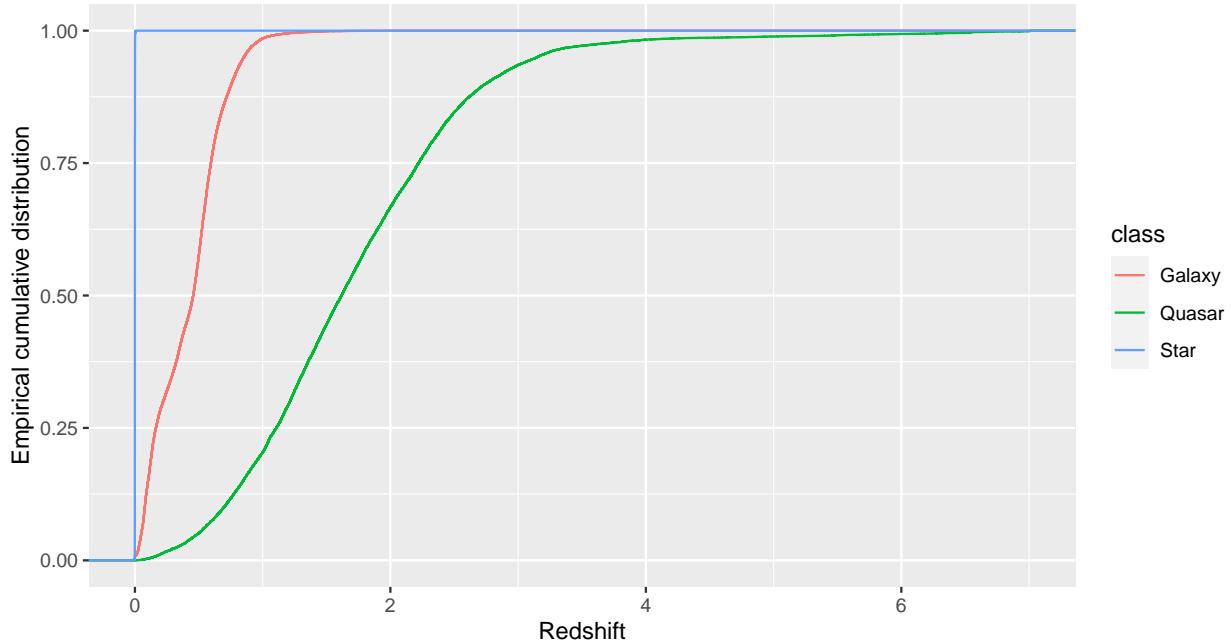
```
ggplot(df, aes(x = `redshift`, y = `class`, fill = ..x..)) +
  geom_density_ridges_gradient(scale = 2, rel_min_height = 1e-5) +
  scale_fill_viridis(name = "Intensity", option = "C") +
  labs(title = 'Redshift distribution per object', x = "Redshift", y = "Object") +
  theme_grey() +
  theme(
    legend.position = "none",
    panel.spacing = unit(0.1, "lines"),
    strip.text.x = element_text(size = 10)
  )
```



Na następnym rysunku podano wykres dystrybuanty empirycznej cechy `redshift` dla każdej klasy obiektów.

```
ggplot(df, aes(x = `redshift`, colour = `class`)) +
  stat_ecdf() +
  theme_grey() +
  labs(title = "Redshift empirical cumulative distribution per object", x = "Redshift", y = "Empirical CDF")
```

Redshift empirical cumulative distribution per object



Chcielibyśmy teraz sprawdzić hipotezę H_0 , że intensywność promieniowania widma podczerwieni gwiazd ma rozkład normalny $N(\bar{z}, S(z))$. Alternatywną hipotezę H_1 określamy następująco: intensywność promieniowania widma podczerwieni nie przyjmuje rozkład normalny $N(\bar{z}, S(z))$.

Dla sprawdzenia hipotezy H_0 tworzymy szereg przedziałowy dla znormalizowanej cechy $\frac{Z-\bar{z}}{s_z}$. Do tabelki również dodajemy kolumnę z odpowiednimi teoretycznymi wartościami rozkładu normalnego.

```
x <- df[df$class == "Star", "z"]
x <- (x - mean(x)) / sd(x)
# szp.n <- ceiling(sqrt(length(x)))
szp.n <- ceiling(log(length(x)))
szp.h <- diff(range(x)) / szp.n
szp <- data.frame(id = 1:szp.n,
                    start = 0:(szp.n - 1) * szp.h + min(x),
                    stop = 1:szp.n * szp.h + min(x))
szp$stop[szp.n] <- szp$stop[szp.n] + szp.h / 10 # to count all the elements
szp[, "mid"] <- apply(szp[, c("start", "stop")], 1, mean)
szp[, "count"] <- apply(szp, 1, function(szp.row) sum(x >= szp.row["start"] & x < szp.row["stop"]))
szp[, "cum.count"] <- cumsum(szp$count)
szp[, "frequency"] <- szp$count / length(x)
szp[, "cum.frequency"] <- cumsum(szp$frequency)
szp[, "theo.frequency"] <- dnorm(szp$mid)
szp[, "theo.count"] <- szp$theo.frequency * length(x)
szp[, "theo.cum.frequency"] <- cumsum(szp$theo.frequency)
szp[, "theo.cum.count"] <- cumsum(szp$theo.count)
round(szp, 2)

##   id start  stop   mid count cum.count frequency cum.frequency theo.frequency
## 1   1 -4.73 -3.82 -4.28     6       6    0.00        0.00        0.00
## 2   2 -3.82 -2.91 -3.36    11      17    0.00        0.00        0.00
## 3   3 -2.91 -2.00 -2.45   295     312    0.01        0.01        0.02
## 4   4 -2.00 -1.08 -1.54  3081    3393    0.14        0.14        0.12
```

```

## 5   5 -1.08 -0.17 -0.63  6106    9499    0.28    0.44    0.33
## 6   6 -0.17  0.74  0.29  6579   16078    0.30    0.74    0.38
## 7   7  0.74  1.65  1.20  4627   20705    0.21    0.96    0.19
## 8   8  1.65  2.57  2.11   851   21556    0.04    1.00    0.04
## 9   9  2.57  3.48  3.02    34   21590    0.00    1.00    0.00
## 10 10  3.48  4.48  3.98     3   21593    0.00    1.00    0.00
##      theo.count theo.cum.frequency theo.cum.count
## 1          0.92           0.00           0.92
## 2         30.10           0.00          31.02
## 3        427.07           0.02          458.09
## 4       2636.05           0.14          3094.14
## 5       7078.75           0.47          10172.89
## 6      8269.95           0.85          18442.84
## 7      4203.32           1.05          22646.16
## 8      929.45            1.09          23575.60
## 9      89.41             1.10          23665.02
## 10     3.12              1.10          23668.14

```

Dla weryfikacji hipotezy H_0 wykorzystujemy test zgodności χ^2 . Przyjmujemy poziom istotności $\alpha = 0.001$.

$$\chi^2 = \sum_{i=1}^r \frac{(n_i - np_i)^2}{np_i}$$

gdzie r – liczba klas, n – liczność próby, n_i – liczność poszczególnych klas, p_i – teoretyczne prawdopodobieństwa należenia do poszczególnych klas. Powyższa statystyka ma rozkład χ^2 z $(r - k - 1)$ stopniami swobody, gdzie k jest liczbą parametrów szacowanych na podstawie próby.

Jak widzimy zmienna losowa χ^2 należy do przedziału krytycznego K , więc należy odrzucić hipotezę H_0 , mówiącą, że intensywność promieniowania widma podczerwieni gwiazd ma rozkład normalny $N(\bar{z}, S(z))$.

```

# chisq.test(szp$count, p = szp$theo.count, rescale.p = TRUE)
chisq <- sum(((szp$count - szp$theo.count)^2) / szp$theo.count)
cat("Chisq =", chisq)

## Chisq = 719.0853
cat("\nCritical interval K: [", qchisq(1 - 0.001, szp.n - 1), ", inf)", sep = "")

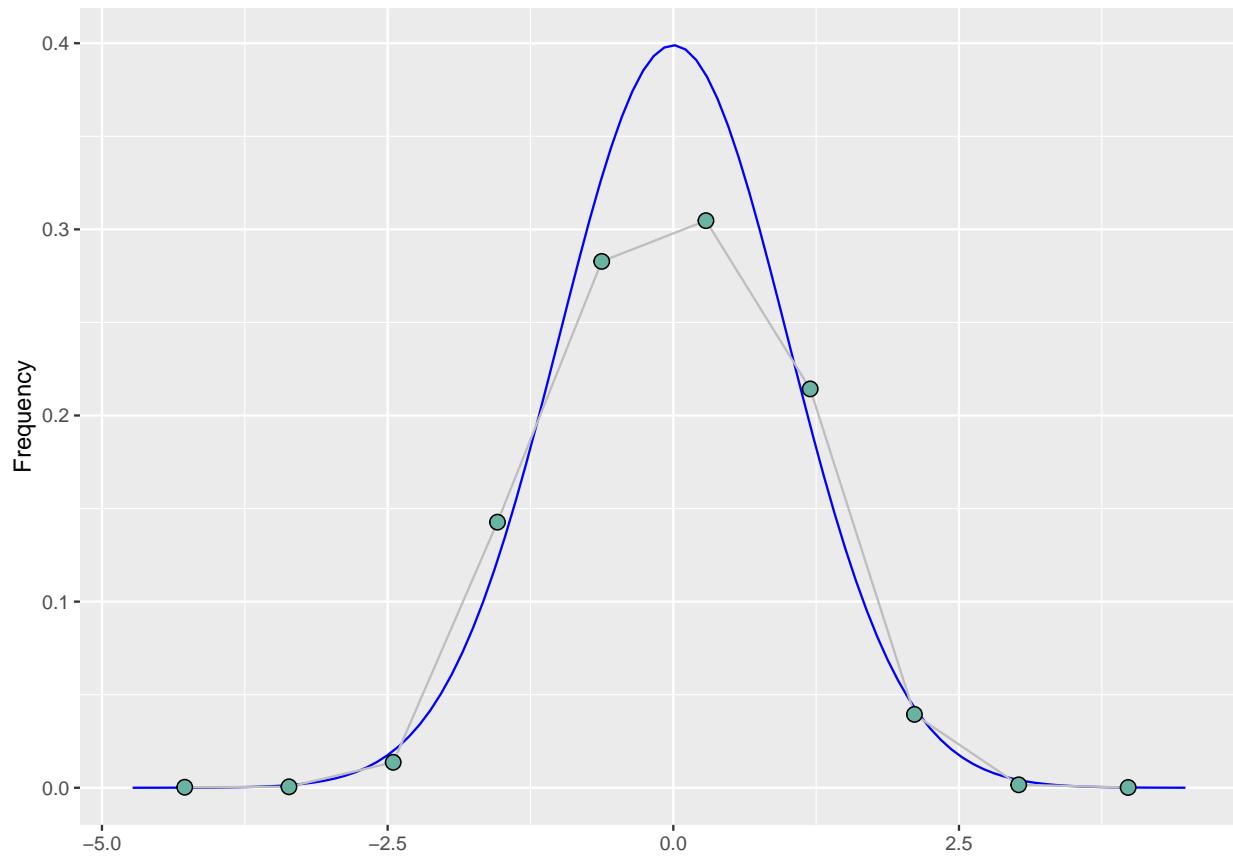
##
## Critical interval K: [27.87716, inf)

Zobaczmy teraz jak wygląda gęstość rozkładu intensywności promieniowania widma podczerwieni gwiazd. Zauważamy istotne różnice w porównaniu do rozkładu normalnego.

x_temp_points <- seq(szp$start[1], szp$stop[szp.n], length.out = 100)
df_norm_dist_plot <- data.frame(xs = x_temp_points, ys = dnorm(x_temp_points))
ggplot(szp, aes(x = mid, y = frequency)) +
  geom_line(data = df_norm_dist_plot, aes(x = xs, y = ys), color = "blue") +
  geom_line(color = "grey") +
  geom_point(shape = 21, color = "black", fill = "#69b3a2", size = 3) +
  labs(title = "Normal distribution vs star infrared distribution", x = "", y = "Frequency") +
  theme_grey()

```

Normal distribution vs star infrared distribution



Eksperyment

Przeprowadźmy teraz mały eksperyment: przyjmujemy, że mamy całą populację intensywności promieniowania ultrafioletowego kwazarów. Teraz w sposób losowy wybieramy z tej populacji próbę, składającą się z 20 elementów. Niech σ_0 - standardowe odchylenie całej populacji. Na podstawie próby 20-elementowej weryfikujemy hipotezę H_0 , która twierdzi, że σ_0 - standardowe odchylenie całej populacji. Powtarzamy taki eksperyment 250 razy dla każdego z następujących poziomów ufności testu: $\{0.80, 0.90, 0.99, 0.999\}$, liczymy ilość testów, w których nie było podstaw do odrzucenia hipotezy H_0 .

W tabelce poniżej podano wyniki takiego eksperymentu. Zauważono, że tylko dla klasy kwazarów wartości praktyczne bardzo różnią się od wartości teoretycznych, dla pozostałych klas różnica jest o wiele mniejsza. Prawdopodobnie wynika to z tego, że rozkład intensywności promieniowania kwazarów znacznie różni się od rozkładu normalnego.

```
y.n.samples <- 250
y.sample.size <- 20
y <- df[df$class == "Quasar", "u"]
sigma0 <- sd(y)
y.sample.V.stats <- c()
for (i in 1:y.n.samples) {
  y.sample <- sample(y, y.sample.size)
  y.sample.V.stats[i] <- y.sample.size * var(y.sample) / sigma0^2
}
df_y_samples <- data.frame(xs = 1:y.n.samples, ys = y.sample.V.stats)
k80 <- c(qchisq(0.2 / 2, y.sample.size - 1), qchisq(1 - 0.2 / 2, y.sample.size - 1))
```

```

k90 <- c(qchisq(0.1 / 2, y.sample.size - 1), qchisq(1 - 0.1 / 2, y.sample.size - 1))
k99 <- c(qchisq(0.01 / 2, y.sample.size - 1), qchisq(1 - 0.01 / 2, y.sample.size - 1))
k999 <- c(qchisq(0.001 / 2, y.sample.size - 1), qchisq(1 - 0.001 / 2, y.sample.size - 1))
df_var_test <- data.frame(a = c("80.0%", "90.0%", "99.0%", "99.9%"),
                           b = paste0(c(
                               sum(k80[1] <= df_y_samples$ys & df_y_samples$ys <= k80[2]) / y.n.samples * 100,
                               sum(k90[1] <= df_y_samples$ys & df_y_samples$ys <= k90[2]) / y.n.samples * 100,
                               sum(k99[1] <= df_y_samples$ys & df_y_samples$ys <= k99[2]) / y.n.samples * 100,
                               sum(k999[1] <= df_y_samples$ys & df_y_samples$ys <= k999[2]) / y.n.samples * 100),
                           "%"),
                           k1 = c(k80[1], k90[1], k99[1], k999[1]),
                           k2 = c(k80[2], k90[2], k99[2], k999[2])))
colnames(df_var_test) <- c("Poziom istotności", "Ile trafiło?", "k1", "k2")
df_var_test

```

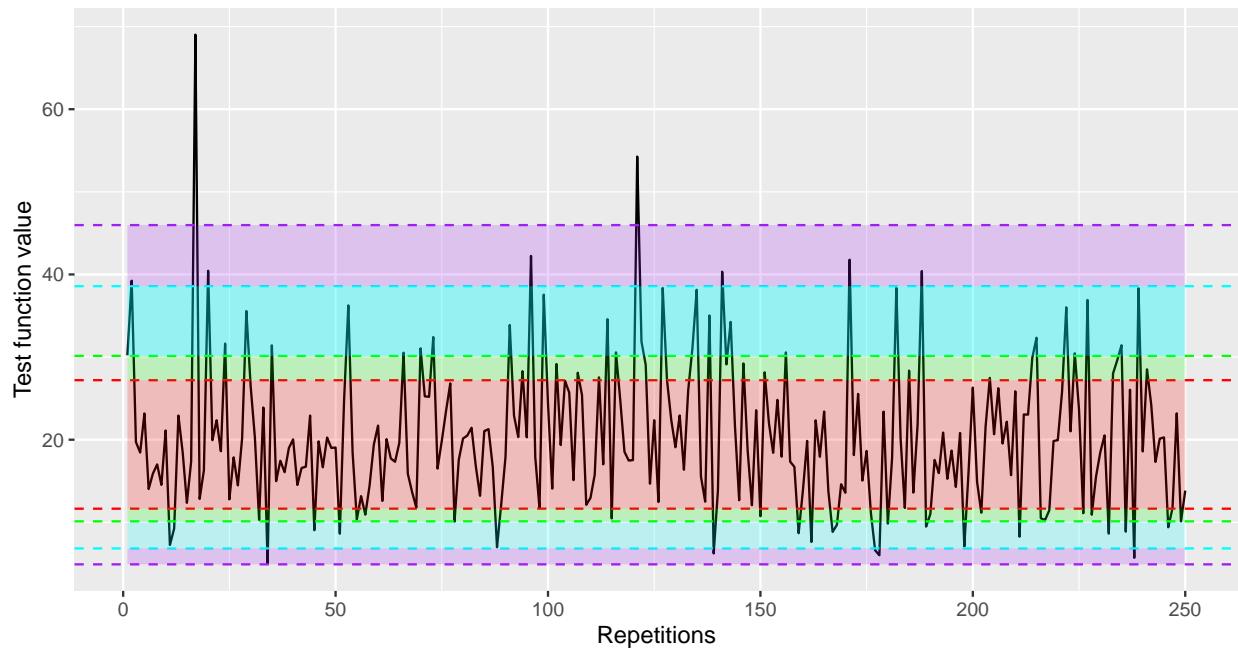
Na poniższym rysunku podano wizualizację przeprowadzonego eksperymentu.

```

ggplot(df_y_samples, aes(x = xs, y = ys)) +
  geom_line(color = "black") +
  geom_hline(yintercept = k80, linetype = "dashed", color = "red") +
  geom_hline(yintercept = k90, linetype = "dashed", color = "green") +
  geom_hline(yintercept = k99, linetype = "dashed", color = "cyan") +
  geom_hline(yintercept = k999, linetype = "dashed", color = "purple") +
  geom_ribbon(aes(ymin = k999[1], ymax = k99[1]), fill = "purple", alpha = 0.2) +
  geom_ribbon(aes(ymin = k99[1], ymax = k90[1]), fill = "cyan", alpha = 0.2) +
  geom_ribbon(aes(ymin = k90[1], ymax = k80[1]), fill = "green", alpha = 0.2) +
  geom_ribbon(aes(ymin = k80[1], ymax = k80[2]), fill = "red", alpha = 0.2) +
  geom_ribbon(aes(ymin = k80[2], ymax = k90[2]), fill = "green", alpha = 0.2) +
  geom_ribbon(aes(ymin = k90[2], ymax = k99[2]), fill = "cyan", alpha = 0.2) +
  geom_ribbon(aes(ymin = k90[2], ymax = k999[2]), fill = "cyan", alpha = 0.2) +
  geom_ribbon(aes(ymin = k999[2], ymax = k999[2]), fill = "purple", alpha = 0.2) +
  labs(title = "Experiment visualization", x = "Repetitions", y = "Test function value")

```

Experiment visualization



```
theme_grey()
```